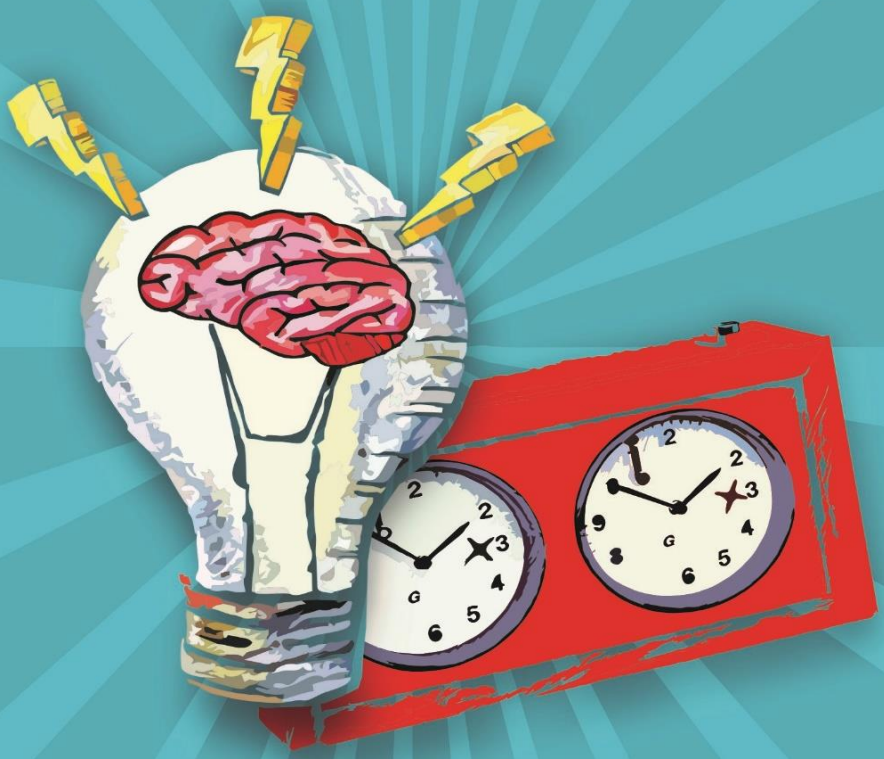
Profesional Ágil

Apuntes para la certificación PMI-ACP®



Pablo Lledó
(PMI-ACP, PMP)

Datos de catalogación bibliográfica

Pablo Lledó

Profesional Ágil: Apuntes para la certificación PMI-ACP®

1a ed. – Estados Unidos: el autor, 2020.

410 p. ; 26x21 cm

ISBN: 9798557870924

1. Negocios. 2. Tecnología.

Editor: Pablo Lledó

Tapa y contratapa: Diego Grifo

Figuras: Paul Leido

Revisión: Sergio García Gómez

Copyright © 2020 Pablo Lledó

Pablolledo es una marca de propiedad de Pablo Lledó

Versión 1.3

PMI, PMP y PMI-ACP son marcas registradas por el Project Management Institute, Inc.

Este libro no puede ser reproducido total ni parcialmente en ninguna forma, ni por ningún medio o procedimiento. Cualquier reproducción sin el permiso previo por escrito de su autor viola los derechos reservados, es ilegal y constituye un delito. Cada copia pdf de este libro es para uso individual.

La disrupción tecnológica, inteligencia artificial, trabajo remoto y los equipos virtuales han cambiado la forma de gestionar nuestros proyectos. Por su parte, la nueva economía está indicando que las empresas exitosas serán sólo aquellas que hagan realidad su estrategia mediante proyectos eficientes que generen los beneficios esperados.

En este contexto ya no alcanza con conocer herramientas para gestionar proyectos de ciclos predictivos con procesos en cascada, sino que cualquier profesional de proyectos debe conocer también cuáles son las metodologías y buenas prácticas para gestionar proyectos con ciclos adaptativos que requieren entregar valor al cliente de manera incremental y con mayor celeridad que en el pasado.

Cuando algunos colegas me solicitaron escribir este libro, mi primera reacción fue "*zapatero a su zapato*". O sea, un Economista no tiene conocimientos técnicos para explicar proyectos de ciclos iterativos adaptativos relacionados con el desarrollo de software.

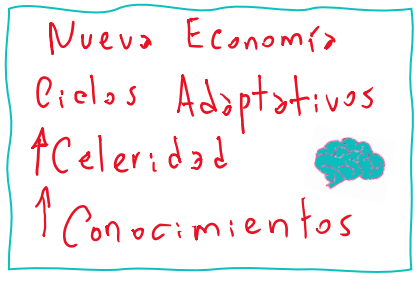
Sin embargo, luego de gestionar algunos proyectos ágiles en mis negocios que no están relacionados con el desarrollo de software, profundicé mis estudios sobre las tendencias de metodologías ágiles.

La forma más fácil de comprender las distintas prácticas fue preparando la certificación PMI-ACP®.

Luego de aprobar esta certificación internacional, aprendí sobre el mundo apasionante de las metodologías ágiles que aplican a cualquier tipo de proyecto. Además, verifiqué que es imposible ser un profesional de proyectos exitoso sin conocer las herramientas de la nueva economía de los proyectos.

Por lo tanto, acá estoy escribiendo este libro en formato de "*apuntes de clase que utilicé para aprobar en el primer intento*" para todos aquellos que quieran preparar la certificación del PMI-ACP® o necesiten mejorar sus conocimientos para la dirección de proyectos.

Estos "apuntes", implican que el libro no tiene introducción-nudo-desenlace en cada capítulo, ni marcos teóricos detallados. Simplemente conocerán la mayoría de los conceptos que les podrían tomar en la certificación PMI-ACP® para aprobar sin morir en el intento o para ser mejores directores de proyecto.



Pablo Lledó es Agile Certified Practitioner (PMI-ACP), Project Management Professional (PMP), Master of Science in Project Analysis (University of York, Inglaterra), MBA en Dirección de Proyectos (Universidad Francisco de Vitoria, España), MBA en Negocios Internacionales (Universitat de Lleida, España) y Licenciado en Economía (Universidad Nacional de Cuyo, Argentina).

Pablo, fundador de varias empresas en marcha y profesor de prestigiosas Universidades, es autor de más de 10 libros sobre Gestión de Proyectos. Ha sido seleccionado como expositor en congresos internacionales del Project Management Institute y ganó el *"PMI Distinguished Contribution Award"*. Actualmente es Director de Pablolledo.com LLC, empresa especializada en Project Management.

Las ventajas de estudiar de este libro son:

- Obtener una visión integral de los principios ágiles más utilizados
- Interactuar con miembros del equipo de proyectos ágiles
- Quedar a poca distancia de aprobar la certificación PMI-ACP®
- Ser mejores profesionales de proyectos ✓✓✓

¿Para qué comprar este libro si lo puedo conseguir gratis?

Debido a la piratería informática y a la falta de ética profesional, es probable que este libro llegara a tus manos sin que lo hayas comprado. Te recordamos que eso es ilegal, ya que la versión digital de este libro es para uso individual.

Te invitamos a invertir sólo $5 dólares comprando este libro en www.pablolledo.com y así seguir alimentando tu buena conducta profesional.

A cambio de tu inversión, Pablo Lledó estará a tu disposición para contestar personalmente cualquier tipo de consulta sobre el libro u otro tema relacionado con la dirección de proyectos. Podrás escribirle a pl@pablolledo.com colocando tu código de compra en el asunto.

Profesional Ágil

Apuntes para la certificación PMI-ACP®

Contenidos

Profesional Ágil 3
Contenidos 5
Introducción 9
Examen PMI-ACP® 9
Materiales de estudio 16
El día del examen 17
Capítulo 1 19
Mentalidad Ágil 19
1.1 Predictivo vs Adaptativo 19
1.2 Manifiesto Ágil 24
1.3 Marco Ágil 27
1.4 Lean 31
1.5 Kanban 34
1.6 Scrum 38
1.7 ScrumBan 45
1.8 Extreme Programming (XP) 46
1.9 Feature Driven Development (FDD) 49
1.10 Dynamic System Development Method (DSDM) 51
1.11 Agile Unified Process (AUP) 53
1.12 Crystal 55
1.13 Scrum of Scrums 57
1.14 Large Scale Scrum (LeSS) 59
1.15 Scaled Agile Framework (SAFe) 61
1.16 Disciplined Agile (DA) 64
1.17 Agilidad organizacional 68
1.18 Ejercicios – Mentalidad Ágil 71
1.19 Respuesta ejercicios – Mentalidad Ágil 91
Capítulo 2 111
Entrega orientada al valor 111
2.1 Entregas incrementales 111
2.2 Priorización de los requisitos 114
2.3 Técnicas de Priorización 121
2.4 Especificación de los requisitos 127
2.5 Entregas rápidas y flexibles 128
2.6 Trabajo en progreso (WIP) 131
2.7 Tiempo de entrega y ciclo 136
2.8 Entregables de calidad 143

2.9 Contratos ágiles 151
2.10 Ejercicios – Valor 153
2.11 Respuesta ejercicios – Valor 164
Capítulo 3 175
Involucrar a los interesados 175
3.1 Interesados 175
3.2 Visión compartida 181
3.3 Comunicaciones 189
3.4 Colaboración 196
3.5 Habilidades blandas 206
3.6 Toma de decisiones 210
3.7 Ejercicios – Interesados 214
3.8 Respuesta ejercicios –Interesados 227
Capítulo 4 239
Desempeño del equipo 239
4.1 Roles del equipo 239
4.2 Organización del equipo 244
4.3 Desarrollo del equipo 250
4.4 Seguimiento del equipo 256
4.5 Ejercicios – Equipo 269
4.6 Respuesta ejercicios – Equipo 278
Capítulo 5 287
Planificación adaptativa 287
5.1 Planificación ágil 287
5.2 Adaptabilidad 289
5.3 Temas, Épicas e Historias 292
5.4 Técnicas de estimación 297
5.5 Planificar iteraciones 308
5.6 Ejercicios – Planificación 311
5.7 Respuesta ejercicios – Planificación 320
Capítulo 6 331
Resolución de problemas 331
6.1 Riesgos 331
6.2 Detectar problemas 338
6.3 Resolver problemas 343
6.4 Seguimiento y control de problemas 348
6.5 Ejercicios – Problemas 353
6.6 Respuesta ejercicios - Problemas 362
Capítulo 7 371
Mejora continua 371
7.1 Retroalimentación 371
7.2 Mejorar los procesos 374
7.3 Mejorar el producto 380

7.4 Mejorar el equipo 385
7.5 Ejercicios – Mejora continua 395
7.6 Respuesta ejercicios – Mejora continua 401
Tips de examen **407**

Introducción

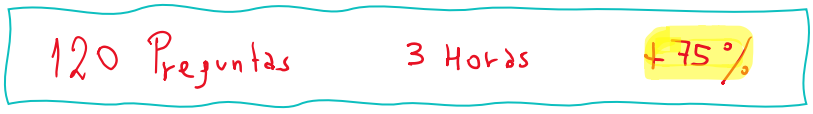
Examen PMI-ACP® [1]

El examen PMI-ACP® consiste en 120 preguntas para rendir en un máximo de 3 horas. 100 preguntas serán puntuadas y 20 preguntas no recibirán puntaje por ser de evaluación. Sin embargo, las preguntas no puntuadas no estarán identificadas en el examen, por lo que el alumno debería rendir pensando en que las 120 preguntas son puntuadas.

El puntaje mínimo para aprobar es variable, pero por lo general se aprueba con 75%. El examen lo puedes solicitar para rendir en español y tendrás la opción de ver la traducción de cada pregunta al inglés en cualquier momento durante el examen.

Las 120 preguntas estarán distribuidas entre 7 dominios.

Dominio	% de Preguntas	Cantidad de preguntas
I. Mentalidad y Principios de Ágil	16%	19
II. Entrega Orientada al Valor	20%	24
III. Involucramiento de los Interesados	17%	20
IV. Desempeño del Equipo	16%	19
V. Planificación Adaptativa	12%	15
VI. Detección y Resolución de Problemas	10%	12
VII. Mejora Continua	9%	11
TOTAL	**100%**	**120**



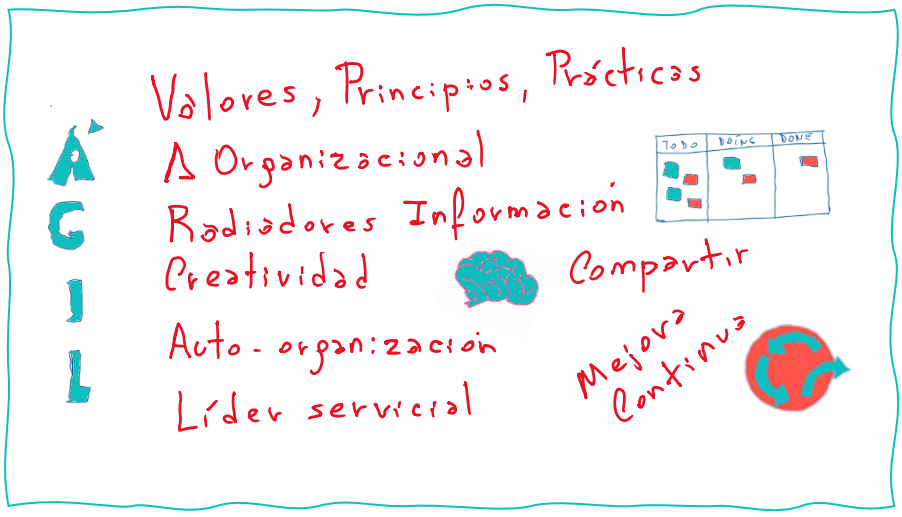
[1] Esquema del Contenido del Examen de PMI Agile Certified Practitioner, Project Management Institute.

A continuación, se presenta un resumen de las tareas que se incluyen en cada Dominio.

Dominio I. Mentalidad y Principios de Ágil

Explorar, adoptar y aplicar la mentalidad y los principios de Ágil en la organización.

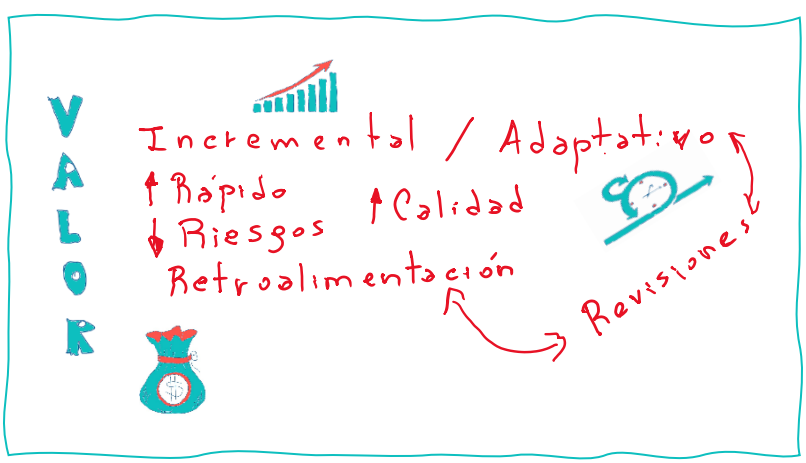
1. Desarrollar una mentalidad de los valores y principios de Ágil en el equipo y cliente.
2. Asegurar que todos comprendan los valores, principios, prácticas y terminología de Ágil.
3. Apoyar el cambio organizacional con procesos y comportamientos más eficientes.
4. Mantener radiadores de información visibles con el progreso real del equipo.
5. Crear un ambiente seguro y confiable para permitir errores para la mejora continua.
6. Mejorar la creatividad experimentando con nuevas técnicas e ideas.
7. Alentar al equipo a compartir conocimientos colaborando y trabajando juntos.
8. Alentar el liderazgo dentro del equipo mediante la autoorganización y la autonomía.
9. Practicar el liderazgo servicial apoyando y alentando a otras personas en sus esfuerzos.



Dominio II. Entrega Orientada al Valor

Entregar valor en forma temprana y frecuente en función de las prioridades de los interesados, obteniendo retroalimentación para mejorar los incrementos futuros.

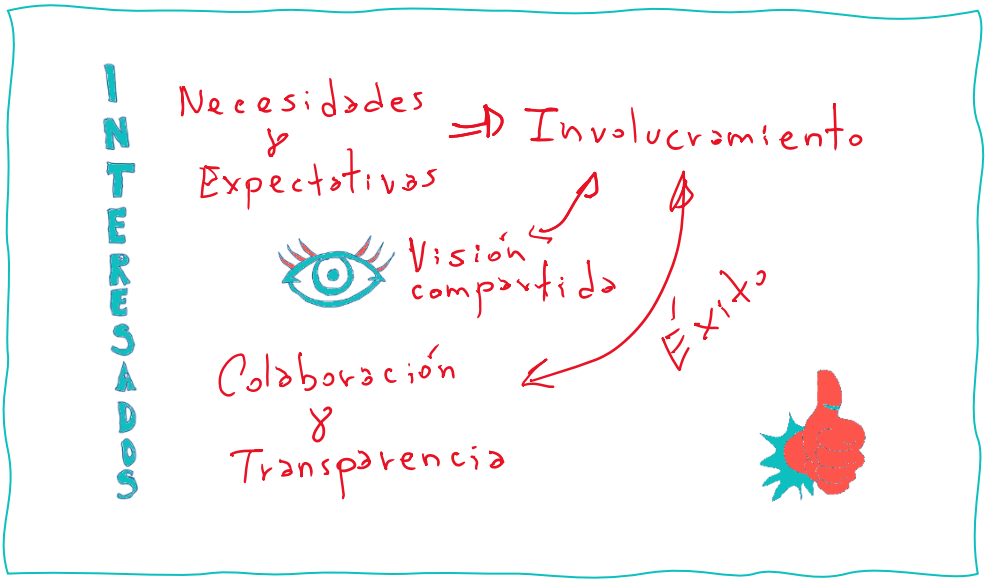
1. Definir los entregables incrementales que maximizan el valor a los interesados.
2. Refinar los requerimientos y sus criterios de aceptación con un enfoque justo a tiempo.
3. Adaptar el proceso según las características del proyecto y la experiencia del equipo.
4. Planificar en función de incrementos pequeños para entregar valor en forma temprana.
5. Realizar revisiones frecuentes con los interesados para mitigar riesgos tempranamente.
6. Solicitar retroalimentación al cliente y a los usuarios para mejorar el valor del negocio.
7. Priorizar el trabajo colaborando con los interesados para optimizar el valor.
8. Realizar mantenimientos frecuentes de la calidad para reducir el costo del desarrollo.
9. Identificar continuamente los factores externos e internos para mejorar la calidad.
10. Realizar revisiones periódicas con los interesados para corregir el trabajo en progreso.
11. Equilibrar el desarrollo de entregables y los esfuerzos de reducción de riesgos.
12. Re-priorizar los requerimientos según cambios en el ambiente e interesados.
13. Priorizar los requerimientos no funcionales según el ambiente en el que se usará.
14. Realizar revisiones frecuentes de los productos para incorporar mejoras en el proceso.



Dominio III. Involucramiento de los Interesados

Involucrar a los interesados creando un ambiente que se alinee con sus necesidades y expectativas, y permita equilibrar sus requerimientos con el costo-beneficio involucrado.

1. Involucrar a los interesados para que el equipo conozca sus necesidades y expectativas.
2. Promover el intercambio de información y conocimiento entre los interesados.
3. Establecer acuerdos entre los interesados claves para promover la colaboración.
4. Mantener un involucramiento adecuado de los interesados evaluando los cambios.
5. Promover la colaboración entre los interesados resolviendo conflictos en forma grupal.
6. Establecer una visión compartida de los entregables del proyecto entre los interesados.
7. Mantener un entendimiento compartido entre los interesados de los criterios de éxito.
8. Proporcionar transparencia sobre el estado del proyecto para tomar decisiones.
9. Proporcionar pronósticos detallados para planificar efectivamente.



Profesional Ágil – Pablo Lledó ©

Dominio IV. Desempeño del Equipo

Crear un ambiente de confianza, aprendizaje, colaboración y resolución de conflictos promoviendo la autoorganización del equipo, las buenas relaciones y alto rendimiento.

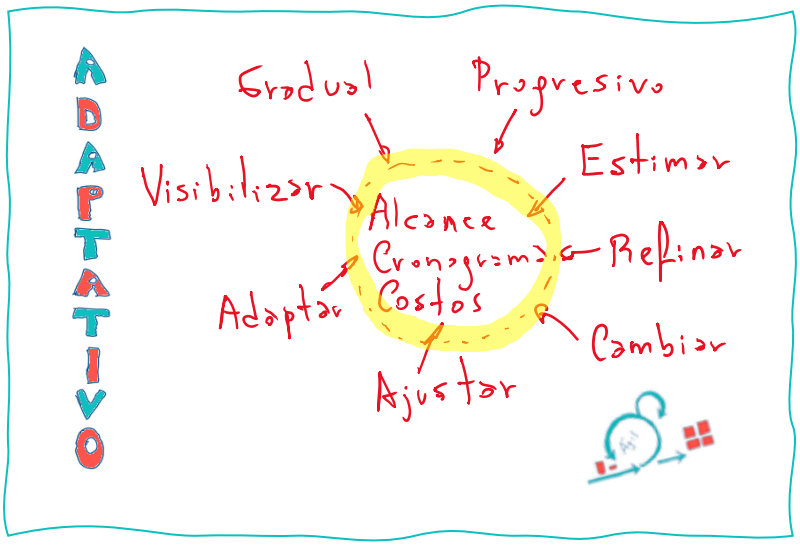
1. Idear reglas básicas a fin de favorecer la cohesión y compromiso del equipo.
2. Desarrollar un equipo con las habilidades necesarias para alcanzar los objetivos.
3. Alentar especialistas generalistas para equipos pequeños de alto rendimiento.
4. Contribuir a la autoorganización del trabajo fomentando el liderazgo emergente.
5. Descubrir continuamente los factores de motivación personales y del equipo.
6. Facilitar la comunicación estrecha mediante la coubicación u otras herramientas.
7. Reducir las distracciones para optimizar el valor entregado.
8. Alinear las metas del proyecto y del equipo compartiendo la visión del proyecto.
9. Alentar al equipo a medir su velocidad monitoreando el rendimiento real.

Profesional Ágil – Pablo Lledó ©

Dominio V. Planificación Adaptativa

Producir y mantener un plan que evoluciona, desde el inicio hasta el cierre, según las metas, valores, riesgos, limitaciones, retroalimentación y hallazgos de las revisiones.

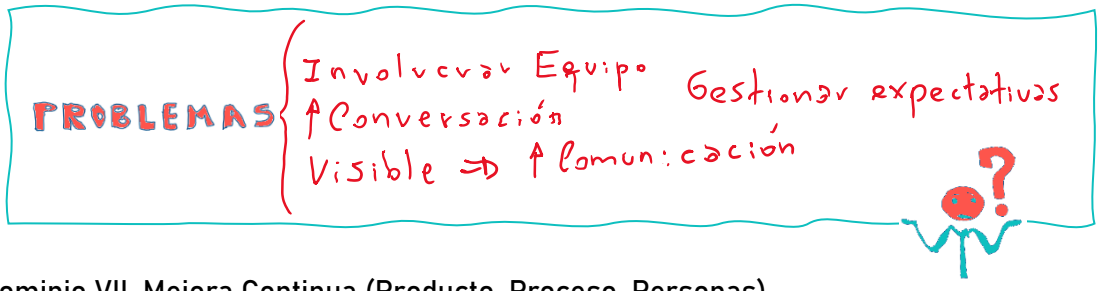
1. Crear el detalle apropiado usando la planificación gradual y elaboración progresiva.
2. Visibilizar y dar transparencia a las actividades de planificación.
3. Gestionar a los interesados garantizando un entendimiento común de los entregables.
4. Adaptar la planificación en función de resultados de retrospectivas periódicas.
5. Inspeccionar y adaptar el plan del proyecto para reflejar los cambios.
6. Utilizar técnicas de elaboración progresiva para determinar el tamaño del proyecto.
7. Ajustar la capacidad incorporando las exigencias de mantenimiento y de operaciones.
8. Crear el alcance inicial, el cronograma y las estimaciones de costo de alto nivel.
9. Refinar el alcance, el cronograma y las estimaciones de costo.
10. Usar continuamente datos de los cambios para estimar la fecha final del proyecto.



Dominio VI. Detección y Resolución de Problemas

Identificar continuamente problemas y riesgos; resolverlos; comunicar el estado de resolución de los problemas; e implementar mejoras para prevenir que vuelvan a ocurrir.

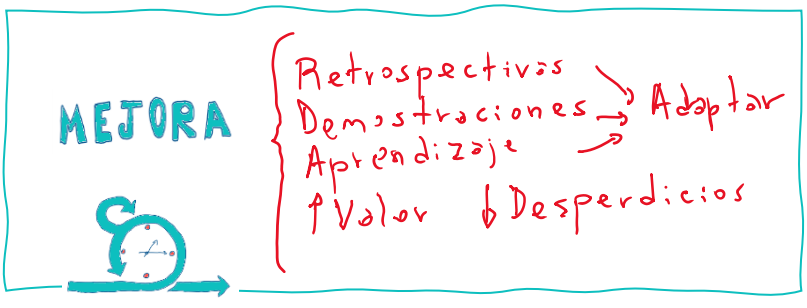
1. Fomentar la conversación y la experimentación para identificar los problemas.
2. Involucrar al equipo para identificar incidentes y resolverlos en el momento apropiado.
3. Garantizar que los incidentes sean resueltos por miembros del equipo adecuados y/o restablecer las expectativas cuando no pueden resolverse.
4. Mantener una lista visible, supervisada y priorizada de amenazas e incidentes.
5. Comunicar el estado de las amenazas e incidentes manteniendo una lista de amenazas.



Dominio VII. Mejora Continua (Producto, Proceso, Personas)

Mejorar continuamente la calidad, efectividad y el valor del producto, proceso y equipo.

1. Adaptar y acondicionar el proceso del proyecto.
2. Mejorar los procesos del equipo llevando a cabo retrospectivas y experimentos.
3. Retroalimentación del producto mediante entregas incrementales y demostraciones.
4. Aprendizaje continuo para desarrollar un equipo de especialistas generalistas.
5. Realizar un análisis de flujo de valor y eliminar desperdicios.
6. Crear mejoras sistémicas distribuyendo conocimiento y prácticas.



Materiales de estudio

Si estás leyendo este libro para aprender sobre conceptos ágiles, deberías complementar tus estudios con los siguientes libros recomendados por el PMI®:

- Agile Estimating and Planning, Mike Cohn
- Agile Project Management with Scrum, Ken Schwaber
- Agile Project Management: Creating Innovative Products, Jim Highsmith
- Agile Retrospectives: Making Good Teams Great, E. Derby, D. Larsen, Ken Schwaber
- Agile Software Development: The Cooperative Game, Alistair Cockburn
- Becoming Agile: ...in an imperfect world, Greg Smith, Ahmed Sidky
- Coaching Agile Teams, Lyssa Adkins
- Guía Práctica de Ágil, Project Management Institute
- Lean-Agile Software Development, Alan Shalloway, Guy Beaver, James R. Trott
- The Art of Agile Development, James Shore
- The Software Project Manager's Bridge to Agility, Michele Sliger, Stacia Broderick
- User Stories Applied: For Agile Software Development, Mike Cohn

Las notas de este libro se basan principalmente en esos libros y en fuentes complementarias como SAFe® , Scrum Alliance y Wikipedia, entre otras.

Si tu objetivo es aprobar la certificación PMI-ACP®, este manual lo deberías complementar con preguntas de simulación. En www.pablolledo.com podrás adquirir simuladores complementarios de este libro.

El día del examen

Todos los detalles de cómo registrarte para el examen, cuándo rendir, cuánto cuesta, etc., los puedes ver en el PMI-ACP® Handbook que puedes descargar gratuitamente desde www.pmi.org.

Por su parte, el examen lo puedes rendir de manera online en la plataforma de Pearson Vue.

En esta sección quiero transmitirte mi experiencia del día del examen, con temas que no están escritos en ningún instructivo.

Si decides rendir desde la computadora de tu casa, que suele ser la opción más práctica, tendrás que estar en un lugar totalmente despejado de personas, libros, apuntes, etc.

Tendrás tu computadora conectada con la webcam y un instructor conectado de manera remota que te estará vigilando todo el tiempo. Te harán apagar completamente el celular y alejarlo del lugar de estudio, deberás mostrar con la webcam debajo del escritorio, debajo de la pc y en todos los rincones de la habitación. No podrás tener reloj de muñeca, ni calculadora y ni siquiera un lápiz y un papel para hacer cálculos.

Podrás tomar notas digitales y usar la calculadora desde la misma plataforma.

No podrás tener tu notebook conectada a un monitor externo.

Si durante el examen cambiara alguna de estas condiciones de "ambiente limpio", o si ingresa alguien en la sala que estás rindiendo, o si te levantaras del monitor, podrían anular tu examen.

Realmente fue una experiencia que no había tenido antes y demuestra los excelentes niveles de monitoreo y control del sistema Pearson Vue.

Una vez que finalices de rendir te saldrá en pantalla la calificación de Aprobado (Above target o Target) o Desaprobado (Below target or Needs improvement). No tendrás una calificación detallada con el porcentaje de respuestas correctas que obtuviste.

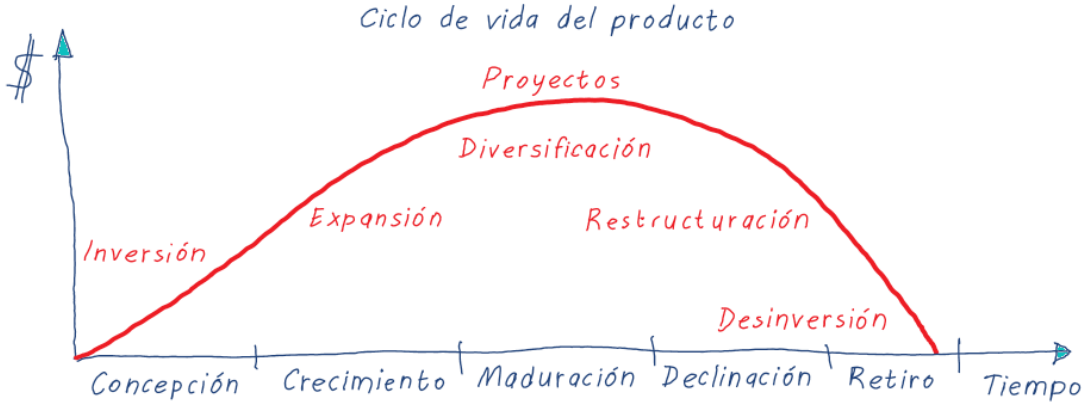
Ni se te ocurra sacar el celular para tomar una foto de esa pantalla hermosa con tu calificación de "Aprobado" , porque el tutor remoto seguirá conectado y podrían anular tu examen. Más adelante te enviarán tu certificado de aprobado por email.

Capítulo 1
Mentalidad Ágil

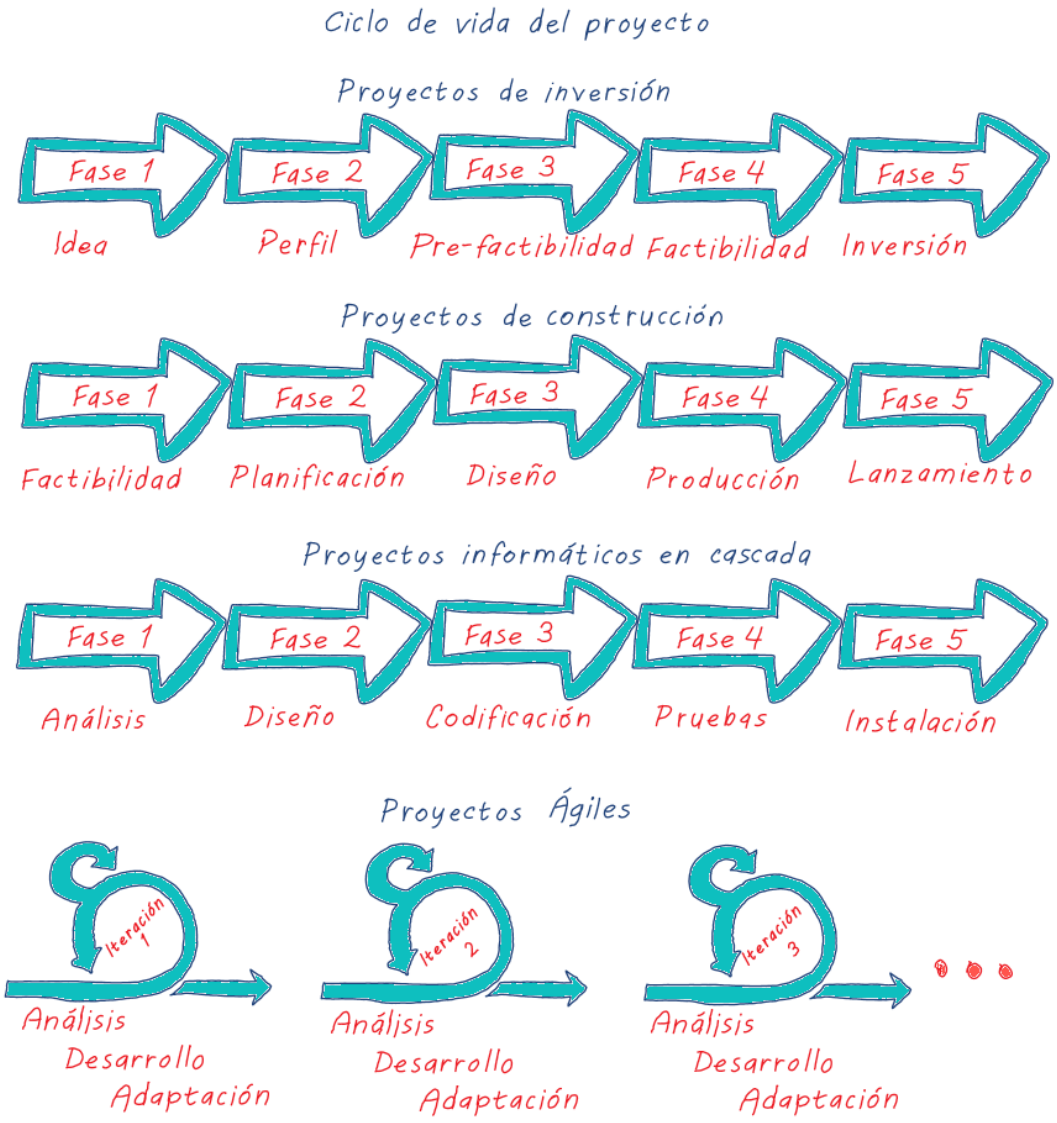
1.1 Predictivo vs Adaptativo

No debemos confundir ciclo de vida del proyecto con el ciclo de vida de un producto. El ciclo de vida del producto es el tiempo que transcurre desde la concepción del producto hasta su retiro del mercado.

Generalmente a lo largo del ciclo de vida de un producto se originan distintos tipos de proyectos como se esquematiza en el gráfico a continuación.



El ciclo de vida del proyecto se refiere a las distintas fases del proyecto desde su inicio hasta su fin. En el gráfico a continuación podemos ver distintos ejemplos de fases de proyectos.

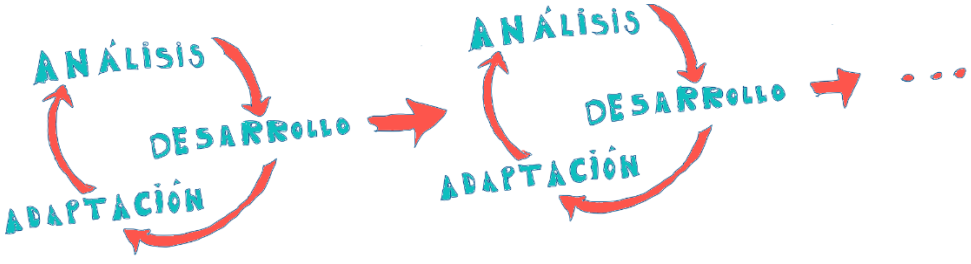


Cada fase del proyecto por lo general termina con un entregable o lección aprendida que habilita o no a continuar con la siguiente fase. Por ejemplo, en un proyecto de ciclo predictivo con fases secuenciales en cascada, si el patrocinador no aprueba el estudio de factibilidad, no podremos comenzar con la siguiente fase de inversión.

Por lo general existen dos tipos de interrelación entre las fases de un proyecto:

- Predictivo: hasta que no finaliza o está avanzada la fase predecesora, no comienza su sucesora. Este ciclo de vida en "cascadas" consiste en seguir un plan desde el inicio hasta el cierre del proyecto. En estos casos, el alcance, tiempo y costo están bien definidos en las fases iniciales del proyecto (inicio, planificación). El ciclo predictivo se suele utilizar cuando se puede definir el alcance al inicio, la frecuencia en las entregas del bien o servicio es relativamente baja y los cambios del mercado son bajos.

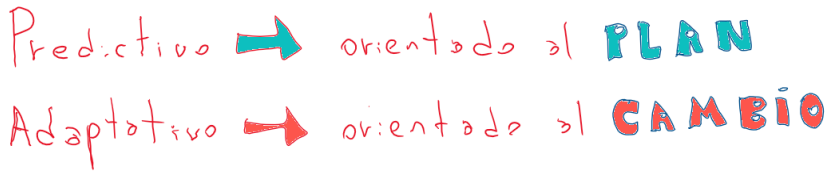
- Adaptativo: se subdivide el proyecto en iteraciones de tiempo fijo (ej. 2 semanas) y cada iteración es gestionada como un mini proyecto. Antes de comenzar con cada iteración, el alcance de esa iteración está definido. Al final de cada iteración se entrega valor al cliente con incrementos parciales del producto o servicio. Antes de comenzar con una nueva iteración, el cliente prioriza el alcance (funcionalidades) de los entregables de esa iteración.



Existen tres variaciones del **ciclo adaptativo**:

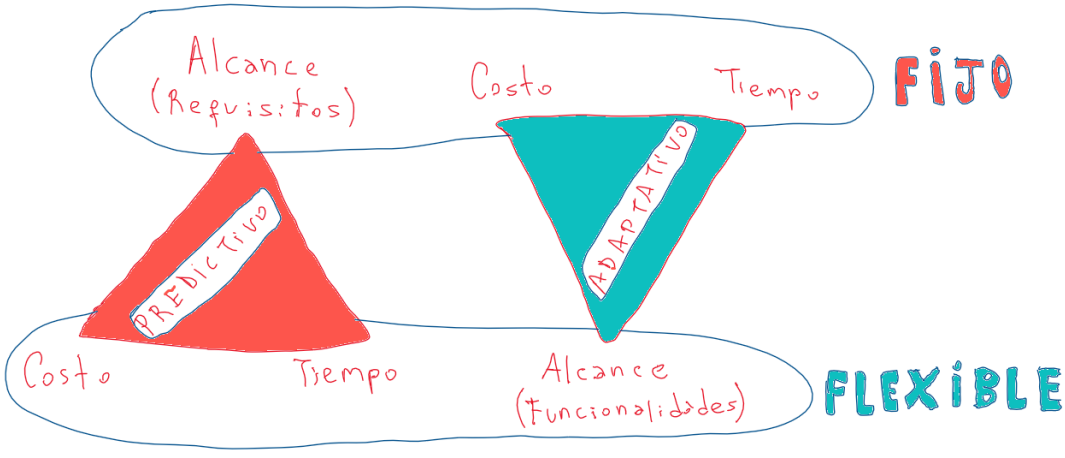
- ✓ **Incremental**: al inicio hay una idea completa sobre el alcance del producto final. En las primeras iteraciones se entrega una funcionalidad básica y se va agregando mayor funcionalidad al producto a medida que avanzan las fases del proyecto. Los entregables de cada fase son un subconjunto de la solución completa y pueden ser utilizados inmediatamente por el cliente. De esta forma se optimiza la velocidad entregando valor al cliente de manera frecuente en mercados con cambios relativamente bajos.
- ✓ **Iterativo**: el alcance preliminar se establece de manera temprana, mientras que el tiempo y costo de cada fase se va definiendo con iteraciones a medida que avanza la ejecución del proyecto. En las primeras iteraciones se va construyendo un borrador del producto final mediante el análisis-desarrollo-reflexión y en las fases sucesivas se va agregando calidad al producto con más análisis-desarrollo-reflexión. Al comenzar no hay certeza sobre el alcance detallado del producto final, sino que se va construyendo a medida que se avanza y se va viendo el desarrollo del producto. Esta forma de trabajo puede llevar más tiempo debido a que está más enfocada en el aprendizaje y adaptabilidad a los cambios del mercado, en lugar de la velocidad en la entrega.
- ✓ **Ágil**: combina ciclos iterativos e incrementales, realizando iteraciones sobre un producto para obtener entregables intermedios listos para usar en cada lanzamiento a lo largo del ciclo del proyecto. Este tipo de interrelación es muy utilizado cuando la frecuencia de las entregas y la incertidumbre del mercado son altas. El principal objetivo no es la velocidad, sino agregar valor para el cliente mediante entregas frecuentes y retroalimentación. Hay diferentes enfoques que utilizan metodologías ágiles como XP, Scrum, Crystal, etc.

También podemos encontrar ciclos de vida de proyectos con una interrelación "**híbrida**" que utiliza fases predictivas para los componentes conocidos del proyecto y fases adaptativas para los componentes inciertos que requieren de mayor aprendizaje y cambios para mejora continua.



En los proyectos de ciclos predictivos por lo general se fija el alcance (requisitos). Luego, el costo y tiempo tendrán que adaptarse en función de ese alcance.

Por otro lado, en los proyectos de ciclos adaptativos en cada iteración se fija el tiempo (ej. entre 15 y 60 días) y costo (ej. número de personas). Luego, el alcance (número de funcionalidades) es la variable que tendrá que adaptarse, por lo que es necesario priorizar las funcionalidades antes de comenzar con cada iteración.



Corrupción del alcance: cambios no controlados en el alcance de un proyecto. En proyectos de ciclo predictivo esto suele ocurrir cuando el alcance no se define, documenta o controla correctamente y se sigue agregando alcance adicional sin pasar por el control integrado de cambios. Por otro lado, en un marco ágil con proyectos de ciclos adaptativos, la corrupción del alcance es poco común ya que la metodología permite cambiar o agregar alcance a lo largo de todo el proyecto. Sin embargo, se recomienda no agregar nuevo alcance en medio de las iteraciones, sino esperar hasta la próxima iteración antes de modificar el alcance.

1.2 Manifiesto Ágil [2]

En el 2001 un grupo de informáticos se reunieron en Utah porque enfrentaban problemas para desarrollar software utilizando las prácticas de ciclos predictivos que existían hasta ese momento. Como resultado de esa reunión redactaron el Manifiesto Ágil:

MANIFIESTO ÁGIL

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones *sobre procesos y herramientas*
Software funcionando *sobre documentación extensiva*
Colaboración con el cliente *sobre negociación contractual*
Respuesta ante el cambio *sobre seguir un plan*

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Individuos e interacciones: prestando más atención a conformar un buen equipo y las relaciones humanas entre esas personas, obtendremos mejores resultados que si implementamos todos los procesos y herramientas que dicen los manuales.

Software funcionando: en lugar de escribir varias páginas explicando los estados de avance de un proyecto o cuándo lo haremos funcionar; es preferible dedicar los escasos recursos a hacerlo funcionar.

Colaboración con el cliente: comprendiendo que nuestro cliente es un aliado estratégico con quien hay que mantener una relación fluida de colaboración, los proyectos terminarán más rápido y con mayor valor con relación a destinar escasos recursos a disputas contractuales.

Respuestas ante el cambio: el plan perfecto no existe, el contexto cambia por lo que tenemos que ser flexibles para adaptarnos a los cambios que necesita el cliente.

[2] http://agilemanifesto.org/iso/es/

Los cuatro valores detrás del Manifiesto Ágil tienen sus **12 principios**:

1 Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

2 Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

3 Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

4 Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

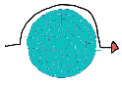
5 Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan y confiarles la ejecución del trabajo.

6 El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

7 El software funcionando es la medida principal de progreso.

8 Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

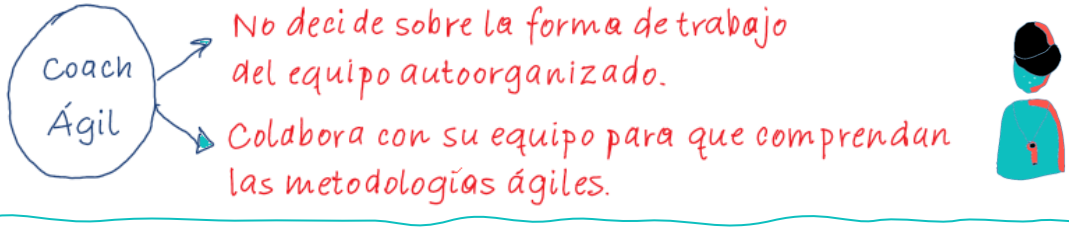
9 La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.



10 La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

11 Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

12 A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.


Coach Ágil
No decide sobre la forma de trabajo del equipo autoorganizado.
Colabora con su equipo para que comprendan las metodologías ágiles.

Declaración de la Interdependencia

En el año 2005, un grupo encabezado por dos de los creadores del Manifiesto Ágil, Alistair Cockburn y Jim Highsmith, escribieron la Declaración de la Interdependencia como una redefinición del Manifiesto Ágil, pero centrado en la gestión de proyectos de software. Luego, el nombre fue cambiado a "*La declaración de interdependencia para la gestión moderna*", ya que los principios aplican a la dirección de proyectos en general.

1 *Incrementamos el retorno de la inversión (ROI) centrándonos en un flujo continuo de valor.*

Entregar valor al cliente lo antes posible. Por ejemplo, es preferible iteraciones de 2 semanas que de 3.

2 *Entregamos resultados fiables haciendo participes a los clientes en interacciones frecuentes y compartiendo la propiedad.*

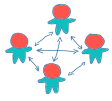
El cliente forma parte activa del desarrollo, participando en la definición del producto y en su revisión frecuente para adaptarlo a sus necesidades.

3 *Esperamos lo inesperado y lo gestionamos mediante iteraciones, anticipación y adaptación.*

El contexto cambia permanentemente, por lo que el trabajo debe ser flexible con ciclos cortos para entregar valor en estas condiciones.

4 *Damos rienda suelta a la creatividad y la innovación reconociendo que los individuos son la fuente última de valor, creando un entorno en el que puedan diferenciarse.*

Cuando los desarrolladores disfrutan lo que hacen en un ambiente distendido, crean bienes y servicios innovadores.



5 *Disparamos el rendimiento mediante la responsabilidad común sobre los resultados y sobre la propia efectividad del equipo.*

Al equipo se les dan las herramientas para que ellos mismos hagan mejor su trabajo y su motivación es causal de la eficiencia.

6 *Mejoramos la efectividad y la fiabilidad mediante estrategias, procesos y prácticas especificas para cada situación.*

No hay una metodología que funcione para todos los entornos. Hay que probar, medir y adaptar hasta encontrar el sistema que mejor funcione.

1.3 Marco Ágil [3]

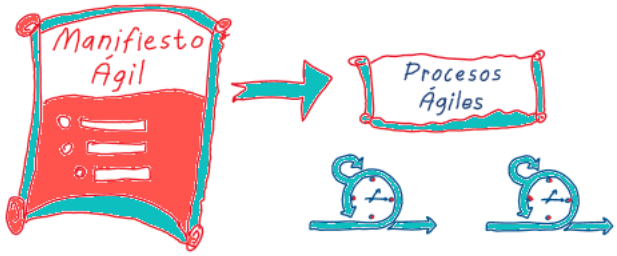
Si bien el Manifiesto Ágil comienza diciendo que las personas son más importantes que los procesos, esto no significa que los proyectos ágiles no necesiten procesos. Los procesos tradicionales en cascada de ciclos predictivos han tenido una mala reputación en las metodologías ágiles por ser estáticos, predictivos y difíciles de cambiar. Sin embargo, todo proceso vinculado con los objetivos estratégicos de la organización que ayude a agregar valor al cliente es bueno.

Por ejemplo, para un proyecto de alcance claro en un contexto de pocos cambios, los procesos predictivos podrían ser la mejor herramienta.

Por otro lado, para aquellos proyectos de innovación dónde es necesario ir modelando el alcance a medida que avanza el proyecto, los procesos de metodologías ágiles flexibles y fáciles de adaptar serían los más apropiados.

Un marco de proceso ágil debe incorporar los principios del Manifiesto Ágil y estar alineado con los objetivos estratégicos de la organización para:

- ✓ Apoyar la visión, explorar y adaptar la cultura.
- ✓ Apoyar equipos autoorganizados y auto-disciplinados.
- ✓ Promover confiabilidad y consistencia según el nivel de incertidumbre
- ✓ Ser flexible y fácil de adaptar.
- ✓ Apoyar la visibilidad del proceso.
- ✓ Incorporar el aprendizaje.
- ✓ Incorporar prácticas que respalden cada fase.
- ✓ Proporcionar puntos de control de gestión para su revisión.



[3] Jim Highsmith, An Agile Process Framework.

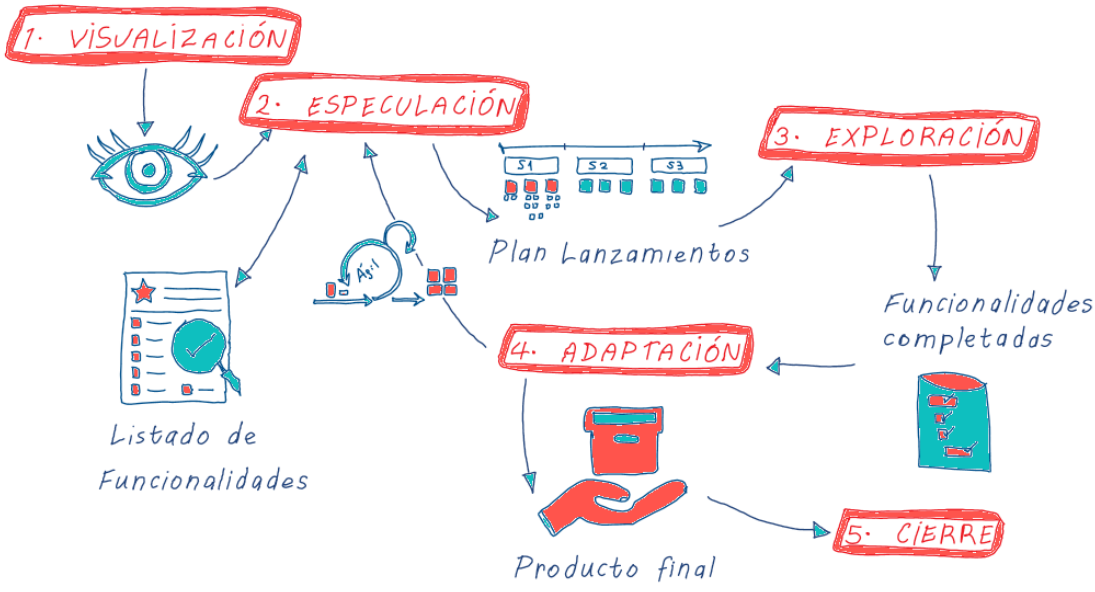
La mayoría de los procesos de metodologías ágiles se centran en la entrega de un bien o servicio mediante iteraciones y la adaptación continua para mejorar el proceso. Las cinco fases del marco ágil desarrolladas por Jim Highsmith, representan bastante bien a la mayoría de las metodologías ágiles.

Fases del Marco Ágil

1. Visualización (Envision)
2. Especulación (Speculate)
3. Exploración (Explore)
4. Adaptación (Adapt)
5. Cierre (Close)

A continuación, se presenta una analogía de la terminología utilizada en ciclos predictivos versus la de ciclos adaptativos.

Predictivo	Adaptativo / Ágil	
Iniciación	Visualización	La visión del proyecto es crítica para el éxito
Planificación	Especulación	Especular indica que el futuro es incierto, en lugar de "plan" que indica predicción y certeza relativa. Hay que especular y adaptar en lugar de planificar y construir.
Ejecución	Exploración	Explorar mediante iteraciones sin procesos en cascada. Requiere flexibilidad porque no se puede predecir completamente los resultados.
Control	Adaptación	Manteniendo el foco en la visión, se monitorea la información y se adapta a las condiciones actuales.
Cierre	Cierre	Transferencia de conocimiento y celebración.



Profesional Ágil – Pablo Lledó ©

1 - Visualización

Determinar la visión del producto y el alcance del proyecto, la comunidad del proyecto y cómo trabajará el equipo en conjunto.

Crear una visión del producto (por qué), una declaración de la misión (cómo), el alcance del producto y sus criterios de éxito (qué), y quiénes serán los principales involucrados. La visualización podría figurar en un documento de una página (visión del producto o acta de constitución) que servirá como la hoja de ruta para alcanzar el éxito.

2 - Especulación

Desarrollar un plan de lanzamientos, hitos e iteraciones necesarias para cumplir con la visualización.

Reunir los requisitos generales iniciales para el producto; definir la lista de características del producto; crear un plan de entregas que incluye cronograma, recursos y costos; incorporar estrategias de mitigación de riesgos; priorizar las funcionalidades a desarrollar en la fase de exploración.

3 - Exploración

Entregar funcionalidades en un corto período de tiempo, buscando constantemente reducir el riesgo y la incertidumbre.

Entregar las características planificadas gestionando el trabajo con las prácticas adecuadas; crear una comunidad colaborativa y autoorganizada facilitada por el líder del proyecto; administrar las interacciones del equipo con los clientes y otros interesados.

4 - Adaptación

Revisar los resultados entregados, la situación actual y el rendimiento del equipo para adaptarlos según sea necesario desde la perspectiva del cliente.

Realizar correcciones, incorporar y retener las lecciones aprendidas para replanificar la próxima iteración.

Después de la fase de Visualización, el ciclo generalmente será Especulación-Exploración-Adaptación, refinando sucesivamente el producto en cada iteración. Sin embargo, podría ser necesario revisar periódicamente la fase de Visualización en función de la nueva información y lecciones aprendidas.

5 - Cierre

Concluir el proyecto, transmitir los aprendizajes y celebrar.

Aprender e incorporar el aprendizaje al siguiente equipo de proyecto.

Planificación Ágil [4]

Quienes piensen que las metodologías ágiles no utilizan demasiada planificación con relación a los marcos de ciclos predictivos, están equivocados.

El marco Ágil tiene planificación y mitigación de riesgos de manera continua e iterativa. En lugar de trabajar con un único plan integral al inicio del proyecto con varios supuestos, trabaja con planificación gradual dónde se detalla lo que se va a realizar en la próxima iteración y se deja a nivel más agregado lo que podría desarrollarse más adelante.

La planificación ágil, también conocida como la **cebolla de planificación**, tiene seis niveles:

Cada capa de la cebolla impulsa los objetivos de las capas siguientes, establece plazos y define la propiedad y la interacción.

La alta gerencia es responsable de:

Estrategia: en qué quiere convertirse la organización en un horizonte de 1 a 5 años.

Portafolio: oferta general de programas, proyectos y productos con sus interrelaciones.

Producto: mapa de ruta del producto de alto nivel que describe cómo se irán logrando los objetivos. (aproximadamente 12 meses)

Los desarrolladores gestionan los tres niveles internos de la planificación:

Lanzamiento: acumulación prioritaria de las funcionalidades del producto que podrían ocurrir aproximadamente cada 3 meses.

Iteración: características planificadas para entregar en incrementos en un plazo corto (entre 2 y 8 semanas). Varias iteraciones terminarán con un lanzamiento.

Día: el equipo planifica qué actividades de la iteración van a realizar durante un día de trabajo. Las pruebas de revisión de calidad se llevan a cabo diariamente durante la iteración.

[4] Mike Cohn, Agile Estimating and Planning.

1.4 Lean

Lean es una filosofía que incluye métodos y herramientas orientados a:

- ✓ Eliminar los gastos y procesos que no agregan valor al cliente
- ✓ Eliminar las demoras e ineficiencias en los procesos de la organización
- ✓ Eliminar fallas, interrupciones y otras pérdidas de producción
- ✓ Buscar de manera continua la perfección y las mejoras de calidad

Principios fundamentales

1. Especificar con precisión el valor de cada proyecto
2. Definir el flujo de valor del proyecto
3. Permitir que el valor fluya sin interrupciones
4. Permitir que el cliente participe en la identificación de "valor"
5. Buscar de manera continua la perfección

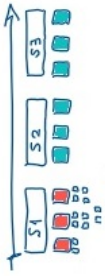
1º - Especificar el valor

"Valor" es cualquier cosa por la que un cliente estaría dispuesto a pagar. Cualquier actividad que no incremente el precio que pagaría el cliente, sólo agrega costos al proyecto.

Las actividades sin valor pueden caer en dos tipos de categorías:

Desperdicios Tipo 1: actividades parcialmente sin valor. Aunque el cliente no pague por ellas, las deberíamos seguir realizando porque son necesarias para el éxito del proyecto. Por ejemplo, realizar reuniones de retrospectiva para mejorar el trabajo del equipo, aunque el cliente prefiera evitar esas reuniones.

Desperdicios Tipo 2: son actividades que carecen de valor agregado y deberían ser eliminadas. Por ejemplo, acortar el tiempo desperdiciado en reuniones, eliminar procesos burocráticos obsoletos, disminuir errores, etc.



2º - Definir el flujo de valor

Todo conjunto de actividades debería tener un entregable y todo entregable debería tener cliente. Por lo general, los entregables que recibe el cliente a través del tiempo componen el flujo de valor.

El mapa de flujo de valor ayuda a visualizar los pasos desde la creación del producto hasta su entrega al cliente. Este mapeo ayuda con la introspección para entender su negocio actual, así como el análisis y la mejora de procesos para eliminar desperdicios.

Por ejemplo, la hoja de ruta con los lanzamientos del proyecto sería un mapa de flujo de valor.

3º - El valor fluye sin interrupciones

Permitir que el valor llegue al cliente lo más rápido posible eliminando del flujo de valor la mayor cantidad de desperdicios como sea posible.

Los "7 desechos" más comunes que hay que eliminar del flujo de valor son: transporte, inventarios, movimiento, esperas, sobre procesamiento, sobreproducción y defectos.

Para no entorpecer el flujo de valor con demoras, deberíamos decidir en el último momento responsable buscando el punto óptimo de la compensación entre el tiempo disponible para una decisión y la necesidad de completar información o desarrollar un producto. O sea, utilizar un sistema de extracción (pull) y "justo a tiempo".

Por ejemplo, en proyectos de manufactura, solo se producen suficientes productos para satisfacer la demanda exacta de los clientes. Por otro lado, en proyectos ágiles, no se detallan todas las funcionalidades de un producto con demasiada anticipación al lanzamiento, ya que a medida que el proyecto avance es probable que algunas funcionalidades se dejen de lado y aparezcan otras nuevas.

4º - El cliente participa

El equipo de desarrollo debe permitir que el cliente colabore en la identificación de lo que agrega valor. El equipo no debe intentar resolver las necesidades del cliente sin consultarles por sus preferencias.

Recordemos que la tecnología en sí misma no agrega valor, sino lo que esa tecnología es capaz de resolver en favor del cliente. Por lo tanto, no hay que dejar la definición del valor solamente en mano de los desarrolladores.

5º - Buscar la perfección continua

Con la mejora continua podremos evitar la Ley de Entropía que indica que las cosas de este mundo siempre tienden por sí solas a volver a su estado caótico natural. En otras palabras, todo lo que hagamos para introducir una cultura lean, se puede perder si no seguimos con la mejora continua día a día.

Por ejemplo, si hemos implementado reuniones diarias de 15 minutos para los avances del proyecto y no nos esforzamos por mantenerlas, podríamos volver a situaciones ineficientes de reuniones semanales de 3 horas.

Un proyecto "lean" requiere vigilancia constante para mantener y mejorar su desempeño, exige disciplina de equipo, una intolerancia total hacia el desperdicio de recursos y la búsqueda permanente de la perfección.

Este principio está relacionado con la palabra Kaizen que persigue la "mejora continua" con pequeños cambios constantes y sostenibles.

Lean Manufacturing existe en Japón desde mediados del siglo XX y se hizo popular en USA y el resto del mundo a partir de 1990 con libro de Womack, Jones y Roos titulado "*The Machine that Changed the World*".

Varios de los conceptos de la filosofía Lean fueron arrastrados por el pensamiento Ágil que comenzó en el 2001 con el Manifiesto Ágil. Por lo que podríamos decir que Lean engloba a Ágil. Además, Lean también abarca a Kanban que explicaremos en el próximo capítulo.

1.5 Kanban

Kanban significa señal o letrero en japonés. El método consiste en colocar el nombre de las actividades y una breve descripción en tarjetas que se pegan en un tablero. Cada tarjeta o actividad va fluyendo desde la recepción de la orden hasta que el trabajo se encuentra terminado. Estos tableros suelen tener gran visibilidad para el equipo y pueden ser en formato físico o digital.

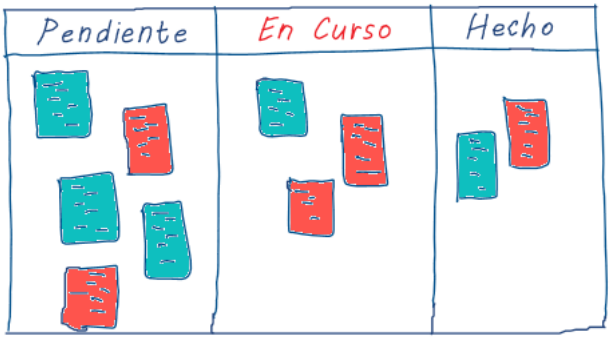
Principios básicos

1. Comenzar aplicando el método en los procesos actuales y luego estimular cambios.

2. Perseguir el cambio incremental, gradual y evolutivo (Kaizen).

3. Respetar el proceso actual, los roles, las responsabilidades y los cargos.

4. Liderazgo en todos los niveles.



Kanban utiliza el enfoque Lean del "sistema pull" y "justo a tiempo". Los miembros del equipo "retiran" (pull) las tarjetas que van a ser procesadas y las trabajan hasta su finalización.

Con los tableros Kanban se pueden coordinar de manera simple las actividades del equipo y sus interrelaciones con otros interesados. El tablero es considerado un radiador de información que no requiere de gran tecnología, puede ser una simple pizarra con post-it dónde los miembros del equipo los van cambiando de casillero.

Los equipos Kanban podrían ser más grandes que lo recomendado por otras metodologías ágiles (ej. 9 personas máximo). El tamaño del equipo dependerá de las tareas a realizar y del número de personas necesarias para acelerar cada entregable con un desperdicio mínimo.

Las cinco **prácticas** centrales del método Kanban son:

1 Visualizar el flujo de trabajo en columnas y hacerlo visible para comprender cómo avanza el trabajo.

2 Limitar el trabajo en progreso (WIP: work in progress) para estimular los cambios.

3 Dirigir y gestionar el flujo de trabajo a través de cada estado.

4 Explicitar las reglas y directrices del trabajo definiendo cuándo y por qué una tarjeta debe pasar de una columna a otra.

5 Reconocer oportunidades de mejora mediante cambios continuos, graduales y evolutivos.

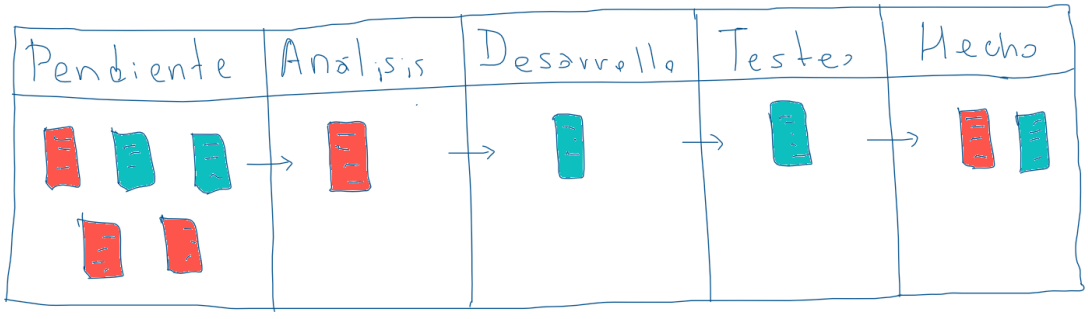
WIP (Work in Progress)

El WIP es la cantidad de tareas en las que un equipo está trabajando actualmente. Kanban gestiona el desarrollo de productos con un enfoque en la entrega continua sin sobrecargar al equipo. Para ello es necesario limitar la cantidad de trabajo en progreso (WIP) para que el equipo no esté ocupado en demasiadas tareas incompletas al mismo tiempo. O sea, es preferible completar rápidamente unas pocas tareas para entregar al cliente en lugar de estar trabajando en muchas tareas incompletas que no agregan valor.

Veamos un ejemplo simple de un equipo que tienen que realizar análisis, desarrollo y testeo a 10 entregables de un proyecto. Para simplificar supongamos que todos los entregables son similares y cada uno de ellos requiere 1 día de trabajo del analista, más 1 día del desarrollador, más 1 día del probador. Si cada miembro del equipo realiza las 10 actividades al mismo tiempo sin priorizaciones, el analista demorará 10 días, cuando termine comenzará el desarrollador con otros 10 días y finalmente el probador demorará 10 días. Los 10 entregables llegarán al cliente en 30 días.

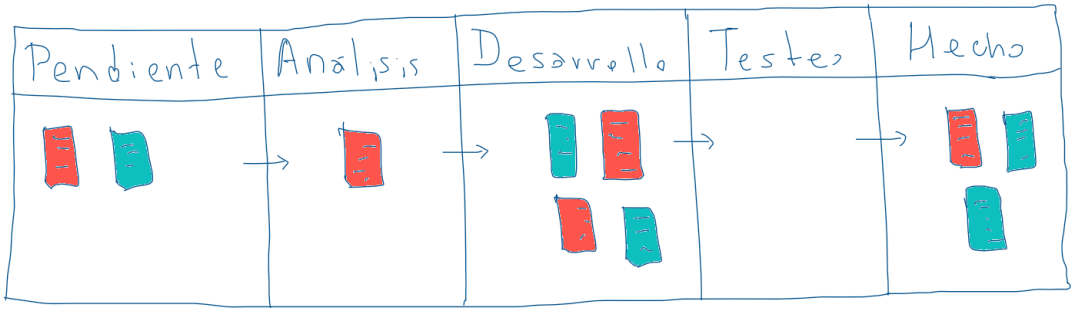
Esta demora podría ser peor debido a las multitareas. O sea, una persona puede demorar 1 día si está enfocado en un entregable a la vez, pero si está realizando 10 entregables al mismo tiempo, las ineficiencias de las multitareas podrían hacer que demore 15 días o más. En este caso, el cliente recibirá los 10 entregables en 45 días como mínimo.

Uno de los principios Lean es que el valor fluya sin interrupciones para que llegue al cliente lo antes posible. Kanban utiliza un tablero gráfico para controlar este flujo. Siguiendo con el ejemplo anterior si se fija el WIP máximo en 1 entregable, el tablero Kanban debería verse de la siguiente forma.



En este caso, el primer entregable llegará al cliente en 3 días (1 día analista + 1 día desarrollo + 1 día testeo). Luego, el cliente seguirá recibiendo un entregable diario porque tanto el analista, el desarrollador y el probador están trabajando con un entregable por día. En 12 días (3 días del primer entregable, más 9 días de los otros 9 entregables) el cliente recibirá los 10 entregables.

Supongamos ahora que el desarrollador se enferma y comienzan a acumularse entregables en su casillero, con el tablero Kanban podemos detectar rápidamente que se ha superado el WIP y se está generando un cuello de botella. Por lo tanto, no tiene sentido comenzar nuevos trabajos pendientes hasta que no se resuelva ese problema. Por ejemplo, se podrían reasignar recursos de “análisis” y “testeo” para colaborar con el “desarrollo” que está trabado.



Cuando estas colas de trabajo o cuellos de botella crecen, los tiempos del ciclo de desarrollo aumentan y la finalización de los entregables se atrasa. Por lo tanto, es más importante completar tareas en proceso que comenzar con tareas nuevas.

Al establecer límites del WIP y utilizar esos límites para guiar cuándo comenzar nuevos productos o servicios, se puede suavizar el flujo de trabajo y reducir los tiempos de entrega, mejorar la calidad y entregar con más frecuencia. Además, al trabajar con lotes pequeños, se facilitan los cambios potenciales minimizando los costos relacionados.

El tablero Kanban también facilita el aprendizaje continuo de todo el equipo al trabajar de manera integrada y permite al equipo que sean ellos quienes tomen las decisiones de "sacar" (pull) las órdenes del flujo de trabajo.

Bucles de retroalimentación

Por lo general existen siete bucles de retroalimentación del sistema Kanban.

	1	trimestral	Revisión de estrategia: definir servicios y el contexto.
	2	mensual	Revisión de operaciones: evaluar el equilibrio entre servicios, personas y recursos para maximizar la entrega de valor.
	3	mensual	Revisión de riesgos: evaluar y responder a los riesgos de entrega en los servicios.
	4	quincenal	Revisión de entrega de servicio: examinar y mejorar la efectividad de un servicio.
	5	semanal	Reunión de reabastecimiento: identificar los elementos en los que trabajará el equipo a continuación.
	6	diaria	Reunión diaria: El equipo coordina sus actividades para el día.
	7	en cada entrega	Reunión de planificación de entrega: monitorear y planificar entregas a clientes.

1.6 Scrum [5]

El término Scrum proviene del rugby donde ocho jugadores con diferentes roles en el juego se entrelazan entre sí para empujar juntos de manera coordinada para ganar la pelota.

Scrum se basa en el empirismo aprendiendo y mejorando de los errores pasados.

Pilares del Scrum

Transparencia**:** dar visibilidad a todo lo que está pasando mediante reuniones de planificación, seguimiento y lecciones aprendidas.

Inspección**:** revisión frecuente de los entregables y del progreso para identificar y corregir las variaciones no deseadas.

Adaptación: hacer los ajustes en los procesos y entregables para la mejora continua.

Conceptos básicos del Scrum

Iterativo-incremental. Entregar al cliente resultados en forma temprana, sobre los cuales él mismo puede hacer ajustes y correcciones.

Equipos automotivados y autogestionados. Cada miembro del equipo toma responsabilidad sobre su trabajo y se compromete con la calidad.

Colaboración constante con el cliente. Se involucra al cliente (dueño del producto) lo largo de todo el proyecto para detectar rápidamente los cambios de requerimientos e incorporarlos en las iteraciones siguientes.

Valores: compromiso, foco, franqueza, respeto y coraje.

[5] Ken Schwaber y Jeff Sutherland, The Scrum Guide.

Ciclo de vida Scrum

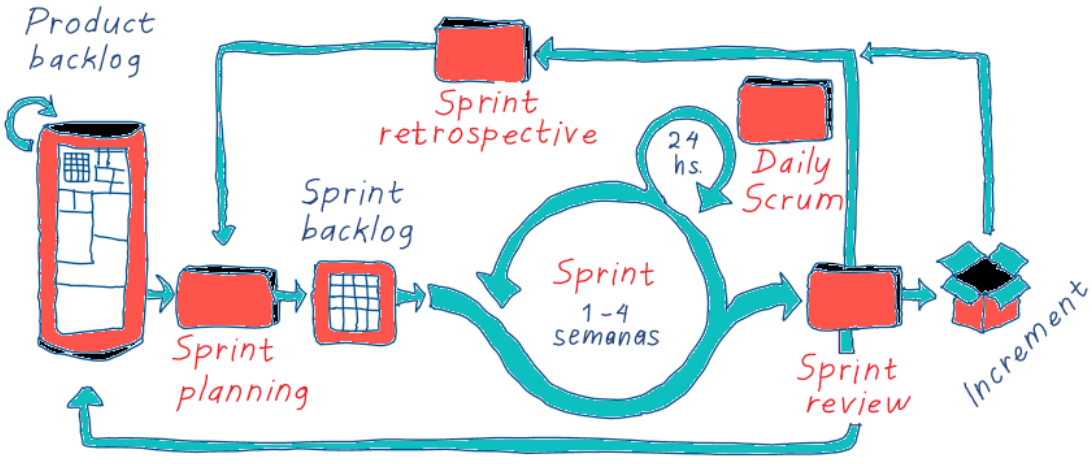
A continuación, se resumen los Artefactos y los Eventos del Scrum.

3 Artefactos

* Trabajo pendiente asociado al producto (Product backlog)
* Trabajo pendiente de la iteración (Sprint backlog)
* Incremento del producto (Increment)

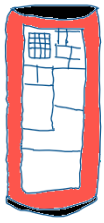
5 Eventos

* Planificación de la iteración (Sprint planning)
* Iteración (Sprint)
* Reunión diaria (Daily Scrum)
* Revisión de la iteración (Sprint review)
* Retrospectiva de la iteración (Sprint retrospective)



Scrum se basa en un ciclo de vida iterativo incremental. Cada iteración se denomina "**sprint**" y tiene una duración fija de entre 1 a 4 semanas. Durante este periodo se realizan los trabajos seleccionados y al finalizar cada sprint se obtiene como resultado un incremento del producto que provee alguna funcionalidad al cliente.

Dentro de un Sprint se podrían crear múltiples incrementos. En estos casos, la sumatoria de todos los incrementos se presentará en la Sprint Review. Para simplificar, en este libro se seguirá mencionando "incremento" en lugar de "incrementos".



Product backlog

El **listado de trabajo pendiente asociado al producto** es un inventario ordenado por prioridades que contiene el trabajo del producto a desarrollar: requerimientos, casos de uso, épicas, historias de usuario, funcionalidades, tareas, etc. Todo lo que hay en el inventario conforma el listado de entregables pendientes que le gustarían al cliente.

Por lo general, el inventario incluye historias de usuario con fichas que describen resumidamente cuál es el requerimiento del cliente o usuario. Los desarrolladores estiman el esfuerzo de cada historia, ya sea en horas o en términos relativos (ej. puntos de historia).

El inventario está disponible para todos los interesados en el proyecto para que contribuyan y aporten sugerencias. Pero el responsable final de establecer las prioridades es el dueño del producto (cliente) en función del valor que le agregan.

El listado de trabajo pendiente es un documento en constante evolución al que se le pueden agregar y quitar historias de usuario en cualquier momento a solicitud del cliente. Los desarrolladores y el dueño del producto realizan un refinamiento continuo del listado de trabajo pendiente agregando detalles y estimaciones a las historias de usuario.

Antes de comenzar cada iteración las prioridades del inventario se revisan y se vuelven a priorizar.

Sprint planning

La **reunión de planificación de la iteración** se realiza antes de comenzar con cada iteración. En esta reunión el equipo planifica de manera colaborativa todo el trabajo necesario para llevar a cabo la próxima iteración. Por lo general es una reunión de 2 horas para cada semana de iteración.

El plan incluye qué parte del producto incremental se podrá completar en la iteración, porqué y cómo se llevará a cabo ese trabajo. El trabajo por realizar se selecciona de las prioridades del product backlog y se coloca en el listado de trabajo pendiente de la iteración.

Durante la planificación se define también el objetivo de la iteración para explicar a los desarrolladores el porqué del producto incremental. Esto permitirá cierta flexibilidad a los desarrolladores sobre las funcionalidades a desarrollar en la iteración. O sea, será más importante cumplir con el objetivo que desarrollar ciertas funcionalidades.

Sprint backlog

El **trabajo pendiente de la iteración** es un plan con el conjunto de elementos del product backlog que han sido seleccionados para completar en la próxima iteración. Incluye las funcionalidades que formarán parte del próximo incremento del producto y todo el trabajo necesario que definieron los desarrolladores para entregar esa funcionalidad.

A partir de la segunda iteración, este trabajo incluye también alguna mejora del proceso identificada en la reunión de retrospectiva.

El plan de trabajo de la iteración debe tener el detalle suficiente para poder evaluar el progreso de las tareas de manera diaria.

Los desarrolladores son quienes deciden el orden de las actividades a realizar y pueden cambiar las prioridades de ese trabajo durante la iteración.

Sprint

Una **iteración** es un período de tiempo fijo (entre una semana y un mes) cuyo objetivo es entregar un incremento del producto que agregue valor al cliente. Cada iteración podría ser considerada como un proyecto.

Las iteraciones tienen siempre la misma duración durante todo el ciclo de vida del proyecto y cada iteración comienza en cuánto termina la iteración previa.

Durante el sprint, cada integrante del equipo toma una historia de usuario y la desarrolla. Cuando finaliza con esa historia, continuará con otra historia de usuario y así sucesivamente hasta finalizar el sprint.

Por lo general, el trabajo pendiente de la iteración no admite cambios ni agregados durante la iteración. Cualquier cambio que surja se reflejará en el product backlog y se podría incorporar en próximas iteraciones según la re-priorización de ese product backlog.

Las ventajas de trabajar con iteraciones de tiempo fijo relativamente bajo es que se acotan los riesgos y costos al periodo de esa iteración, permitiendo aplicar mejoras y cambios en las próximas iteraciones.

De manera diaria se analiza el progreso del trabajo realizado y el trabajo pendiente del Sprint. El trabajo pendiente de la iteración se puede representar gráficamente en el Sprint Burndown Chart.
Cada sprint debería acercar el producto al objetivo del producto.

Daily Scrum

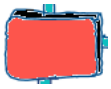
La **reunión diaria** de 15 minutos se realiza por lo general al inicio de la jornada laboral y tiene el objetivo de que los desarrolladores sincronicen las actividades que van a realizar en las próximas 24 horas.

La estructura de la reunión es establecida por los desarrolladores. Mientras algunas se basan en discusiones, otros prefieren enfocarse en preguntas, por ejemplo: ¿Qué trabajo se realizó ayer?, ¿Qué trabajo se va a realizar hoy?, ¿Qué impedimentos se visualizan para realizar el trabajo?

El scrum master asegura que los desarrolladores tengan la reunión, pero los desarrolladores son los responsable de llevarla a cabo.

Los miembros del equipo a menudo se reúnen inmediatamente después de la reunión diaria para discusiones más detalladas, para resolver algún problema, o para replanificar el trabajo pendiente de la iteración.

La reunión diaria es interna para los desarrolladores y el scrum master. Cualquier otro interesado puede asistir a la reunión, pero no deberían interrumpir durante esa reunión.



Sprint Review

La **reunión de revisión de la iteración** por lo general se lleva a cabo al final de cada iteración para inspeccionar el incremento del producto elaborado. Sin embargo, un incremento podría presentarse a los interesados antes de que finalice la iteración.

Los miembros del equipo hacen una demostración del producto a los interesados clave para que realicen una retroalimentación de cómo mejorar el valor del producto. Como resultado de esta retroalimentación se planifica con mejor información la próxima iteración y se adapta el listado de trabajo pendiente del producto si fuera necesario.

Esta reunión informal dura una hora por cada semana de iteración. O sea, para una iteración de 15 días, la reunión será de 2 horas.

Increment

El **incremento del producto** es la sumatoria de todos los elementos del listado de trabajo realizado durante la iteración que se agregaron a los resultados de las iteraciones previas.

Al final de la iteración el incremento debe estar en condiciones de uso. Por lo tanto, el dueño del producto podría decidir hacer un lanzamiento de ese incremento para que esté disponible para el usuario final.

Un incremento completado es el que cumple con la definición de terminado (medidas de calidad requeridas para el producto).

Sprint Retrospective

La **retrospectiva de la iteración** es una reunión donde el equipo scrum se inspeccione a sí mismo para analizar qué funcionó bien y qué aspectos habría que mejorar. Con estas lecciones aprendidas se elabora un plan con las mejoras en los procesos a implementar en la próxima iteración.

Esta reunión ocurre después de la revisión de la iteración donde se presentó el producto incremental a los interesados y antes de la próxima planificación de la iteración.

Suele ser una reunión de 45 minutos por cada semana de iteración. O sea, para una iteración de 2 semanas, la retrospectiva será de 1,5 horas.

Aunque las mejoras pueden implementarse en cualquier momento del proyecto, la retrospectiva brinda una oportunidad formal para centrarse en la inspección, la adaptación y la mejora continua.

Equipo Scrum

El equipo scrum está formado por el dueño del producto, los desarrolladores y el scrum master. Este equipo autogestionado elige la mejor manera de lograr el trabajo sin que estén dirigidos por personas fuera del equipo. Además, el equipo multifuncional tiene todas las competencias para realizar el trabajo sin depender de otros.

Product Owner

El **dueño del producto** es una persona con conocimientos del negocio que representa los intereses del cliente o usuario. Es responsable de maximizar el valor del producto:

- Expresa claramente los elementos del product backlog (objetivos, funcionalidades, características, alcance de las historias de usuario)
- Prioriza los componentes del product backlog en función del valor del producto optimizando sobre el trabajo que realizará el equipo
- Asegura que el product backlog sea visible, transparente y claro para todos, indicando las historias de usuario de la próxima iteración
- Acepta o rechaza el incremento del producto de cada iteración

Los **desarrolladores** son profesionales que realizan el trabajo necesario para entregar el incremento del producto.

El equipo de desarrolladores suele tener un tamaño mínimo de 3 personas hasta un máximo de 8 personas. Suelen trabajar a tiempo completo en el proyecto.

Developers

- Autogestionado. Tienen autoridad suficiente para tomar las decisiones de quiénes y cómo llevarán a cabo el producto.
- Multifuncional. Poseen todas las habilidades necesarias de un equipo interdisciplinario para desarrollar el producto. (Ej. programadores, diseñadores, administrador de base de datos, probadores, etc.)
- Sin títulos. No existen cargos jerárquicos entre sus miembros.
- Responsabilidad compartida. Cada persona puede tener diferentes roles, pero la responsabilidad de los resultados pertenece al equipo.

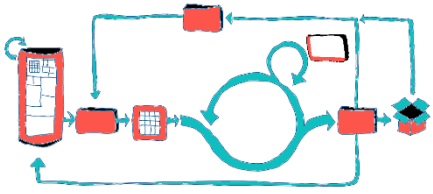
El **scrum master** es un líder servicial responsable de apoyar al equipo a comprender la teoría, las prácticas, las reglas y los valores de Scrum.

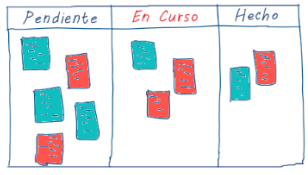
Scrum Master

- Lidera y entrena a la organización en la adopción de Scrum
- Asegura que los objetivos, el alcance y la propiedad del producto sean entendidos por todo el equipo
- Asegura que el dueño del producto sepa cómo organizar y priorizar la cartera de productos para maximizar el valor
- Ayuda a los interesados externos a comprender cuáles de sus interacciones con el equipo son útiles y cuáles no
- Aísla al equipo de interferencias externas y soluciona los problemas que pudieran surgir para asegurar la productividad del equipo
- Entrena al equipo sobre autogestión y multifuncionalidad
- Asegura que el equipo tenga la reunión diaria de 15 minutos
- Facilita las reuniones (eventos de scrum) cuando sea necesario

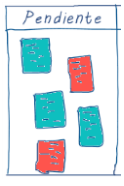
1.7 ScrumBan

Scrumban combina la estructura de iteraciones de Scrum con la visualización de Kanban.



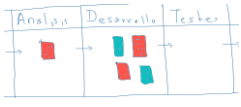


Las reuniones de planificación se llevan a cabo sólo cuando sea necesario determinar qué historias de usuarios se completarán en la próxima iteración. O sea, se planifica bajo demanda lo que permite ahorrar tiempo al equipo en relación con las planificaciones recurrentes de Scrum.

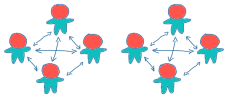


El desencadenante de planificación está asociado con la cantidad de tareas que quedan en la sección "Tareas pendientes" del tablero. Cuando cae por debajo de cierto número, se realiza la planificación para priorizar las tareas a realizar en la próxima iteración.

Para mantener las iteraciones cortas, se mantiene bajo el límite de trabajo en progreso (WIP).



Al igual que en Kanban, el tablero Scrumban revela cuellos de botella en el flujo de trabajo.



Scrumban no requiere ningún número máximo de miembros del equipo o roles específicos como utiliza Scrum. Los roles que tiene un equipo antes de adoptar Scrumban se mantienen.

1.8 Extreme Programming (XP) [6]

Extreme Programming (**XP**) es un marco ágil de desarrollo de software que tiene como objetivo producir software de mayor calidad mejorando al mismo tiempo la calidad de vida del equipo. El método describe prácticas esenciales y alienta a los equipos a utilizarlas.

Valores

Comunicación: transferir el conocimiento entre los miembros del equipo mediante discusiones cara a cara y dibujos.

Sencillez: evitar desperdicios realizando sólo las cosas absolutamente necesarias.

Retroalimentación: recibir comentarios constantes sobre el trabajo para identificar áreas de mejora y revisar las prácticas. La retroalimentación del sistema demora pocos minutos y la del cliente algunos días.

Coraje: plantear problemas que reducen la eficacia de su equipo; dejar de hacer lo que no funciona y probar otra cosa.

Respeto: compartir comentarios que honren la relación del equipo, trabajando juntos para identificar soluciones simples.

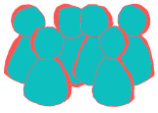
Prácticas

Las 12 prácticas originales de XP son: juego de planificación, pequeños lanzamientos, metáfora, diseño simple, pruebas, refactorización, programación en pareja, propiedad colectiva, integración continua, semana de 40 horas, cliente en el sitio y norma de codificación.

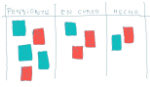
[6] Agile Alliance, https://www.agilealliance.org/

Las prácticas originales han sido refinadas con el correr del tiempo y en la actualidad las más utilizadas suelen ser las siguientes:

Sentarse juntos. Todos los miembros del equipo están colocados en la misma sala, manteniendo zonas para privacidad cuando sea necesario.



Todo el equipo: equipo multifuncional trabajando juntos diariamente en el mismo espacio para lograr un resultado específico.



Espacio de trabajo informativo: radiadores de información para comunicar activamente actualizaciones.

Trabajo energizado: equipo enfocado y libre de distracciones para poder concentrarse en el proyecto. No trabajar demasiado (max. 40hs semanales) y mantenerse saludable, ya que las personas se desempeñan mejor y más creativamente si están bien descansadas. Si hay horas extras una semana, la semana siguiente no debería incluir más horas extras para poder mantener un ritmo sostenible.

Programación en pareja: el software es desarrollado por dos personas sentadas en la misma máquina, obteniendo una revisión continua del código y una respuesta más rápida a los problemas. Esto implica propiedad del código compartido.

Historias de usuarios: descripciones breves de las cosas que los usuarios desean poder hacer con el producto.

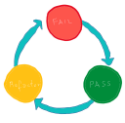
Ciclo semanal: sinónimo de iteración. El equipo se reúne el primer día de la semana para reflexionar sobre el progreso hasta la fecha, el cliente elige las historias que le gustaría completar esa semana y el equipo determina cómo abordarán esas historias.

Ciclo trimestral: sinónimo de lanzamiento. El cliente presenta el plan general al equipo con las funcionalidades deseadas dentro de un trimestre.

Holgura: agregar algunas tareas o historias de baja prioridad en sus ciclos semanales y trimestrales que se pueden eliminar si el equipo se atrasa en historias más importantes.

Construcción de diez minutos: construir automáticamente todo el sistema y ejecutar todas las pruebas en diez minutos para ejecutar con alta frecuencia y no posponer errores.

Integración continua: integrar y confirmar código cada pocas horas, evitando esfuerzos de desarrollo divergentes o fragmentados por falta de comunicación entre los desarrolladores.



Prueba de primera programación (Test-First Programming): seguir el ciclo "Escribir prueba automatizada fallida -> Ejecutar prueba fallida -> desarrollar código para hacer pasar la prueba -> ejecutar prueba -> repetir"; en lugar del ciclo tradicional "desarrollar código -> escribir pruebas -> ejecutar pruebas".

Diseño incremental: comenzar con poco trabajo para comprender la perspectiva adecuada del diseño del sistema, y luego profundizar en los detalles de ese diseño cuando se quiera entregar características específicas. La refactorización (limpiar el código) forma parte del diseño incremental para eliminar la duplicación del código innecesario.

Roles

Cliente. Toma todas las decisiones de negocio relacionadas con el proyecto (qué, cuándo, cuánto, prioridades)

Desarrollador. Realizan las historias priorizadas por el cliente. Son especialistas genéricos que pueden cubrir varios roles (ej. desarrollador, probador y rastreador) para reducir los cuellos de botella y el tamaño del equipo.

Rastreador. Realiza un seguimiento de las métricas de progreso del proyecto (ej. velocidad, horas extras, errores, etc.) para identificar mejoras. Suele ser uno de los desarrolladores que utiliza parte de su tiempo cumpliendo el rol de rastreador.

El entrenador. Consultor externo que ha usado XP anteriormente y guía a los otros miembros del equipo en las prácticas de XP y la autodisciplina.

1.9 Feature Driven Development (FDD)

El desarrollo impulsado por funcionalidades (FDD: Feature Driven Development) se utiliza en el desarrollo de software con equipos grandes más complejos y menos autoorganizados que Scrum u otras metodologías similares. Contempla además el rol del jefe de proyectos.

La metodología comienza con procesos secuenciales para el diseño de arquitectura y luego trabaja con iteraciones de una duración máxima de 15 días para entregar producto tangible que cubra la funcionalidad solicitada por el cliente.

Procesos

1º Desarrollar modelo general: definir la visión, el contexto y los requisitos del sistema. Luego, se crean modelos de dominio detallados por áreas.

2º Desarrollar lista de funcionalidades: elaborar con el cliente el listado de funcionalidades que debe tener cada dominio. Cada funcionalidad es definida con el formato "<acción> <resultado> <objeto>" que puedan ser completadas en un plazo máximo de 2 semanas. Por ejemplo: “Validar la contraseña de un usuario”.

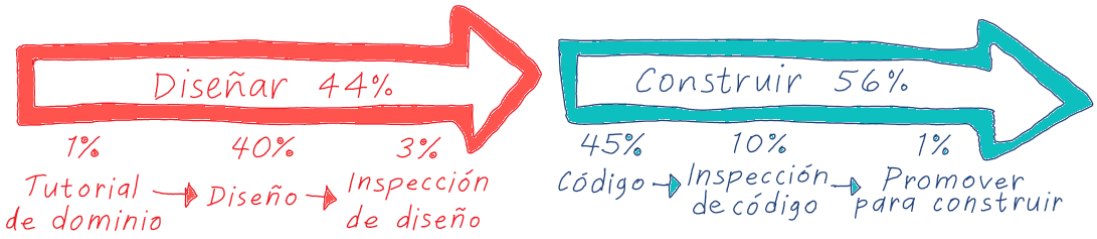
3º Planificar por funcionalidad: ordenar las funcionalidades y asignarlas a los programadores jefe según prioridades y dependencias

4º Diseñar por funcionalidad: el programador jefe decide qué funcionalidad realizar en cada iteración y elabora diagramas de secuencia detallados.

5º Construir por funcionalidad: desarrollar el código, realizar pruebas, inspeccionar y se agrega cada funcionalidad a la compilación general.

Hitos

Para obtener informes de estado y realizar un seguimiento de cada funcionalidad, se definen 6 hitos que se van completando secuencialmente con porcentajes de avance predefinidos.



Prácticas

- Modelado de objetos de dominio. Explorar y explicar el dominio del problema a resolver.
- Desarrollando por funcionalidad. Cualquier función que sea demasiado compleja para implementarse dentro de dos semanas se descompone en funciones más pequeñas.
- Propiedad de código individual. Se asignan distintas piezas de códigos a un solo propietario responsable de la coherencia, el rendimiento y la integridad conceptual.
- Equipos por funcionalidad. Equipos pequeños desarrollan cada funcionalidad. Múltiples mentes evalúan varias opciones de diseño antes de seleccionar una.
- Inspecciones. Detección de defectos para garantizar un diseño y un código de buena calidad.
- Gestión de la configuración. identificar el código fuente de todas las características que van completando y a mantener un historial de cambios.
- Construcciones regulares. Demostraciones frecuentes del producto al cliente para detectar errores de integración de manera temprana.
- Visibilidad de progreso y resultados. Informes de avance frecuentes, apropiados y precisos.

Roles: Director de proyectos, Jefe de arquitectura, Gerente de desarrollo, Jefe de programación, Dueño de clase, Experto de dominio.

1.10 Dynamic System Development Method (DSDM)

El Método de Desarrollo de Sistemas Dinámicos (DSDM) se creó en el Reino Unido en base a prácticas comerciales con el objetivo de brindar funcionalidad a tiempo y dentro del presupuesto.

Cada proyecto se divide en partes, y cada parte tiene un número específico de características, presupuesto y tiempo.

Si un proyecto se está quedando sin tiempo o presupuesto (que son fijos), las características (funcionalidades) menos importantes se descartan y se consideran para proyectos futuros.

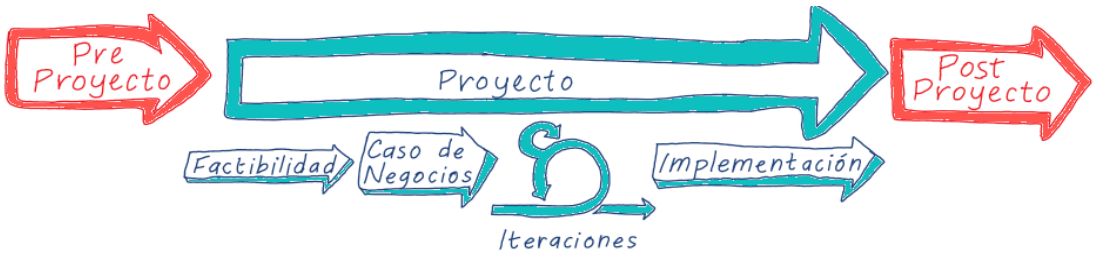
Fases

1º - Pre Proyecto: Cosas que deben ocurrir antes de que comience el proyecto.

2º - Ciclo de vida del proyecto: se produce el proyecto en etapas:

- Estudio de factibilidad
- Caso de negocios
- Iteración del modelo funcional
- Diseñar y construir iteraciones
- Implementación

3º - Fase posterior al proyecto: cosas que deben ocurrir después de que se haya completado el proyecto.

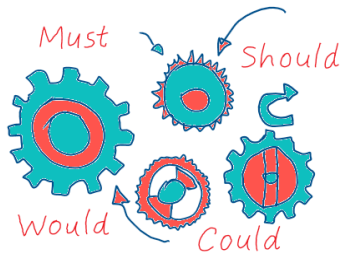


El tamaño máximo del equipo de proyecto suele ser 6 equipos de 6 personas cada uno.

DSDM se centra en la entrega frecuente de productos, asumiendo que entregar algo temprano es mejor que entregar todo al final. Al entregar el producto frecuentemente, se pueden detectar errores de manera temprana para mejorar en la siguiente iteración.

El 80% del beneficio de la empresa proviene del 20% de los requisitos del producto. Por lo tanto, DSDM comienza implementando primero ese 20% de funcionalidades. Implementar la totalidad de requisitos a menudo causa que un proyecto supere plazos y presupuestos, así la mayoría de las veces es innecesaria construir la solución perfecta.

Las características del producto (requisitos) se priorizan según MoSCoW (**M**ust have, **S**hould have, **C**ould have, **W**ould like to have):



- DEBE tener este requisito de manera obligatoria
- DEBERÍA tener este requisito si fuera posible
- PODRÍA tener este requisito si se puede entregar sin mayor impacto
- ME GUSTARÍA agregar estos requisitos si queda suficiente tiempo

Principios

1. Enfocarse en las necesidades del negocio. Entregas cumplen necesidades actuales.
2. Entregar a tiempo. La entrega frecuente es clave.
3. Colaboración. La participación del usuario es imprescindible.
4. Nunca comprometer la calidad. Equipo capacitado para cumplir.
5. Construcción incremental sobre cimientos sólidos.
6. Desarrollo iterativo. Pruebas integradas durante todo el ciclo de vida.
7. Comunicación clara y continua.
8. Demostrar control utilizando las técnicas apropiadas. Los cambios son reversibles.

Características principales

- Aceptación de DSDM por la alta gerencia y empleados para que estén motivados desde el principio y permanezcan involucrados durante todo el proyecto.
- Compromiso de la administración para garantizar la participación del usuario final mediante la creación de prototipos funcionales.
- Equipo compuesto por miembros hábiles empoderados para tomar decisiones sobre el proyecto sin tener que escribir propuestas formales a la alta gerencia.
- Apoyo entre el cliente y el proveedor.
- Retroalimentación de los usuarios sobre los requisitos son cruciales
- Utilizar prototipos para refinar los requisitos y aclarar las expectativas de los interesados.

1.11 Agile Unified Process (AUP)

El Proceso Unificado Ágil

(AUP: Agile Unified Process) es un proceso de desarrollo de software iterativo e incremental; "serial en lo grande" e "iterativo en lo pequeño". Esto significa que las cuatro fases principales son secuenciales y cada una de esas fases se divide en iteraciones.

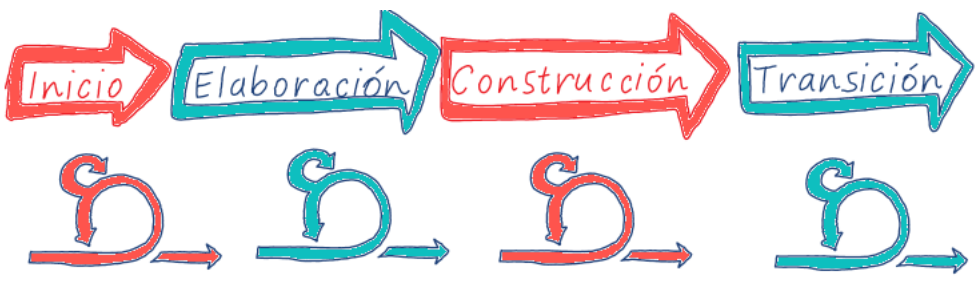
Fases

Inicio: desarrollar la visión compartida del alcance del producto y definir su arquitectura.

Elaboración: ampliar la comprensión de los requisitos del producto y validar la arquitectura.

Construcción: completar el desarrollo del producto.

Transición: completar las pruebas del producto para su lanzamiento.

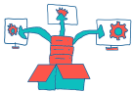


Principios

1. El equipo sabe lo que está haciendo. El producto proporciona enlaces a muchos de los detalles para quienes estén interesados.
2. Simplicidad. Todo se describe de manera concisa en pocas páginas.
3. Agilidad. Se ajusta a los valores y principios del desarrollo de software ágil.
4. Foco en actividades de alto valor. Centrarse en las actividades importantes, no todas las cosas posibles que le pueden pasar en un proyecto.
5. Herramientas independientes. Se puede usar cualquier conjunto de herramientas.
6. Adaptar metodología para el uso. Cada proyecto es único por lo que la metodología debería ser flexible.
7. Específico a cada situación. Ajustar las interrelaciones organizacionales para cada proyecto en particular.

Disciplinas de AUP

AUP es una simplificación de RUP (Rational Unified Process) que utiliza las siguientes 7 disciplinas.

Modelo	Comprender el negocio de la organización y el dominio del problema del proyecto para identificar una solución.
Implementación	Transformar los modelos en código ejecutable y realizar un nivel básico de pruebas unitarias.
Prueba	Realizar una evaluación objetiva para garantizar la calidad. Encontrar defectos, verificar que el producto funciona como se diseñó y validar que se cumplen los requisitos.
 Despliegue	Planificar la entrega del producto y ejecutar el plan para que el producto esté disponible para los usuarios finales.
Gestión de la configuración	Gestionar el acceso a los componentes del proyecto, el seguimiento de las distintas versiones y los cambios.
Gestión del proyecto	Gestionar las actividades del proyecto, personas, entregas a tiempo y dentro del presupuesto, rastrear el progreso y administrar riesgos. Los elementos de alto riesgo se priorizan temprano en el desarrollo.
Ambiente	Asegurarse que el proceso, los estándares y las herramientas estén disponibles para el equipo cuando sea necesario.

En el 2012, AUP fue reemplazada por el modelo de entregas de Disciplined Agile (DA).

1.12 Crystal

Crystal es una familia de metodologías ágil creada por Alistair Cockburn para el desarrollo de software.

Crystal se subdivide en varios tipos de metodologías en función a la cantidad de personas y la criticidad del proyecto.

Propiedades

1. Entregas frecuentes. Ciclo de vida iterativo e incremental.
2. Mejora reflexiva o mejora continua en cada iteración.
3. Comunicación osmótica colocando al equipo en el mismo lugar físico para comunicarse cara a cara.
4. Seguridad personal donde todos pueden expresar su opinión sin miedos porque siempre se considera su opinión.
5. Enfoque. Trabajo sin interrupciones durante 2 horas.
6. Fácil acceso a usuarios expertos. Mínimo semanalmente.
7. Entorno técnico con pruebas automatizadas. Gestión de la configuración e integración continua.

Valores: personas, interacción, comunidad, habilidades, talentos y comunicaciones.

Cuantas más personas tenga el equipo, más crítico será el proyecto y más complejo será el enfoque. A medida que crece el tamaño del equipo (desde 1 hasta 200 personas), la metodología Crystal cambia de color para agregar más formalidad a la estructura, los artefactos y la gestión del proyecto.

Claro

Admite precio fijo, sin contrato de negociación. Orientado a las personas, no se enfoca demasiado en procesos y artefactos. Requiere documentación mínima. Los roles básicos son patrocinador, diseñador y programador. Los miembros del equipo trabajan juntos en la misma habitación. Se trabaja en iteraciones que van entre los 15 a 90 días.

Amarillo

Propiedad clara de las áreas de código, donde el propietario de ese código puede realizar cambios. Los comentarios se obtienen de los "Usuarios reales". Reduce la necesidad de demasiada documentación. Las pruebas automatizadas se utilizan para resolver los errores más rápido. Se establecen planes de mejora mensuales.

Naranja

El proyecto dura de 1 a 2 años. Divide al equipos según sus habilidades funcionales. El número de roles aumenta incluyendo arquitecto, patrocinador, analista de negocios, gerente de proyectos, probador, etc. Se requiere un lanzamiento del incremento del producto cada 3-4 meses. Diseñado para proyectos medianos. Define un conjunto de productos específicos: documento de requisitos, secuencia de lanzamientos, cronograma del proyecto, reportes de estado, documento de diseño, manual de usuario, casos de prueba, etc.

Rojo

Se sigue el método tradicional de desarrollo de software con fases secuenciales. Los equipos se forman y se dividen según el trabajo requerido.

Marrón

Para proyectos de gran tamaño. Los métodos definidos son diferentes según la necesidad del software.

Diamante

Proyectos muy críticos y de gran escala. Su equipo y estrategias se deciden según la criticidad del proyecto. Implican el riesgo potencial de la vida humana.

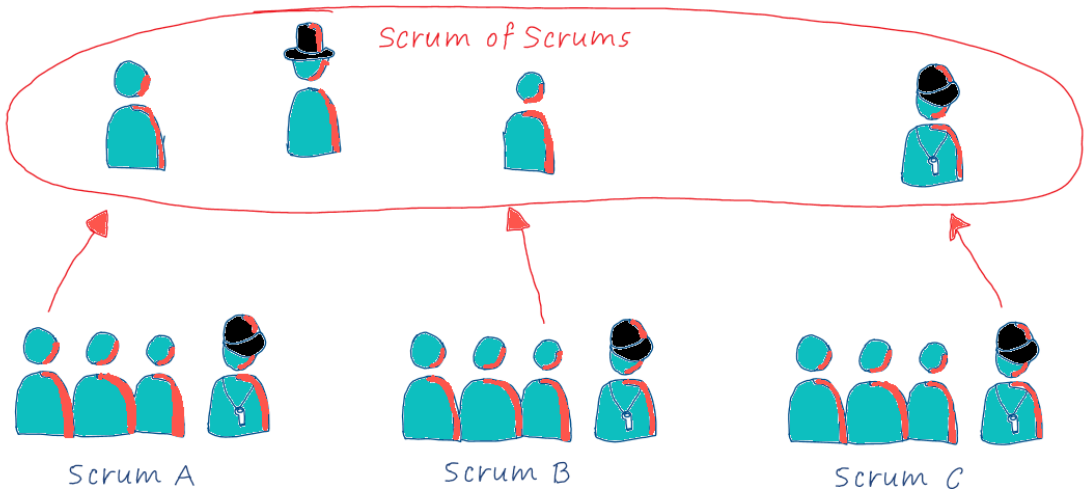
La comodidad (ej. cuántas horas podría trabajar cada recurso), el dinero disponible, el dinero requerido para completar el proyecto y la vida del proyecto, son otros factores que junto con el tamaño del equipo influyen en la selección de una metodología u otra.

1.13 Scrum of Scrums

Scrum por lo general trabaja con equipos de un tamaño máximo de hasta 10 personas (8 desarrolladores, 1 dueño del producto, 1 scrum master). Sin embargo, hay organizaciones que trabajan con proyectos que requieren coordinar esfuerzos de equipos superiores a 10 personas. En este ámbito es donde aparece **Scrum of Scrums**, también denominado **MetaScrum**.

Scrum of Scrums (**SoS**) coordina múltiples equipos Scrum que trabajan sobre el mismo producto, coordinando discusiones sobre el progreso en sus interdependencias y centrándose en cómo integrar la entrega del producto, especialmente en áreas en las que haya superposición.

La forma de coordinar estos esfuerzos entre los diferentes equipos scrum es a través de la reunión SoS que demora entre 15 y 30 minutos. Cada uno de los equipos scrum, envían a un representante del equipo, el embajador, para que participe de esta reunión.



La reunión SoS, o reunión de embajadores, se suele realizar con una frecuencia de una o dos veces por semana. Sin embargo, algunas organizaciones al finalizar la reunión diaria scrum, llevan a cabo la reunión SoS con una frecuencia diaria.

Todos los equipos Scrum coordinados bajo SoS trabajan en el mismo producto. Por lo tanto, comparten un mismo listado de trabajo pendiente (product backlog) y un único dueño del producto. El dueño del producto suele participar de las reuniones de SoS.

El equipo comprometido con SoS está compuesto por miembros individuales de los diversos equipos de desarrollo. El equipo de desarrollo decide qué miembro del equipo debe enviarse al SoS en función de quién puede ser un mejor orador para resolver los problemas de dependencia entre equipos.

La persona que representa a un equipo durante el SoS puede cambiar con el tiempo según quién esté mejor capacitada para representar al equipo en cada momento. En algunos casos, el equipo de desarrollo envía como embajador al scrum master o al coach ágil.

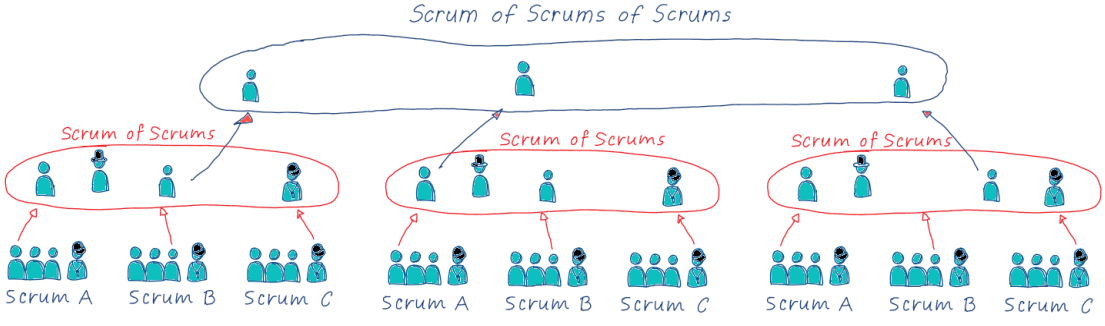
Los participantes en el SoS responden preguntas similares a las del scrum diario:

- ¿Qué ha hecho mi equipo hasta ahora desde la última vez que nos reunimos que podría afectar a otros equipos?
- ¿A qué problema se enfrenta mi equipo que podría obtener ayuda de otro equipo para resolverlo?
- ¿Qué hará mi equipo antes de que nos volvamos a ver que pueda afectar a otros equipos?

Las coordinación de las reuniones SoS suelen ser responsabilidad de los equipos, por lo que no es necesario asignar un coordinador. Sin embargo, algunas organizaciones asignan a un Scrum Master Jefe para la coordinación y el seguimiento de esas reuniones.

La reunión SoS no debería convertirse en una reunión de estado de avance del proyecto donde los scrum masters reportan a la gerencia el progreso de los equipos de desarrollo. El objetivo de SoS es asegurarse de que los equipos individuales logren sus objetivos de la iteración y que se cumpla el objetivo general del proyecto de todos los equipos.

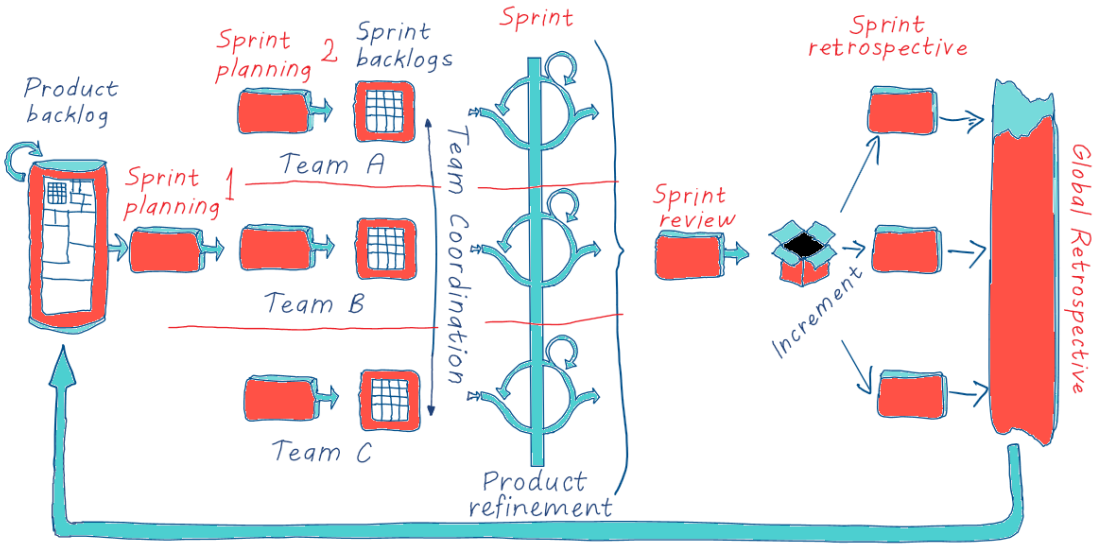
Aquellas organizaciones con proyectos muy grandes suelen aplicar el mismo concepto de SoS a "Scrum of Scrums of Scrums", donde embajadores de cada uno de los SoS participan de una reunión de coordinación de más alto nivel.



1.14 Large Scale Scrum (LeSS) [7]

Large-Scale Scrum (LeSS) es un marco de desarrollo de productos que extiende Scrum cuando es necesario desarrollar un mismo producto con varios equipos de trabajo. La metodología LeSS se utiliza para un máximo de 8 equipos Scrum que tengan hasta 8 personas por equipo.

A continuación, se resumen un ejemplo de LeSS para tres equipos Scrum.



LeSS mantiene los siguientes conceptos básicos de Scrum:

1 product backlog y 1 dueño del producto

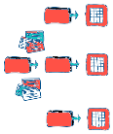
1 incremento del producto al final de cada iteración

Equipos multifuncionales completos en cada equipo

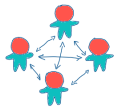
[7] https://less.works/

Los principales agregados de LeSS a Scrum son:

Hay más de un equipo Scrum trabajando en paralelo sobre el mismo producto



La planificación del Sprint se divide en dos partes. Sprint 1: el dueño del producto y representantes de cada equipo coordinan qué parte del sprint backlog realizará cada equipo en la próxima iteración. Sprint 2: cada equipo planifica el detalle de las actividades de la iteración (sprint backlog).



Durante la iteración los diferentes equipos comparten información de lo que están realizando. Por ejemplo, enviando un delegado del equipo a que participe como oyente en las reuniones diarias de los otros equipos o mediante reuniones de Scrum of Scrums.

A mitad de la iteración, los equipos se reúnen con el dueño del producto para hacer un refinamiento del product backlog para aclarar las necesidades de los interesados y asegurarse que el trabajo pendiente sea lo suficientemente pequeño para poder completarlo en lo que queda de la iteración.

Cuando cada equipo finaliza con su retrospectiva de la iteración, todos los equipos coordinan una reunión de retrospectiva global.

Los roles en LeSS son los mismos que en Scrum: un único dueño del producto, el equipo de desarrollo y el scrum master. Cada equipo puede tener su propio scrum master, o un mismo scrum master podría brindar soporte hasta un máximo de tres equipos al mismo tiempo.

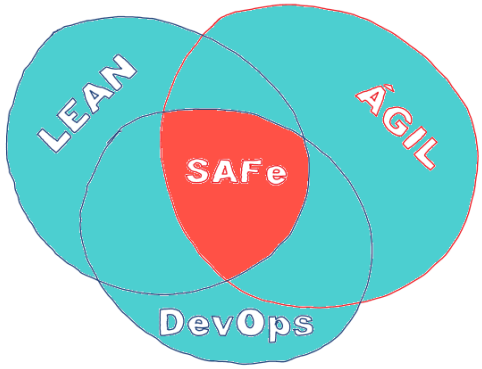
Para proyectos que utilicen más de 8 equipos se utiliza LeSS Huge, donde el product backlog se separa en áreas con representantes del producto para esas áreas.

Enterprise Scrum: [8] marco de referencia para aplicar Scrum a un nivel agregado de toda la organización utilizando enfoques que permiten la innovación disruptiva. El alcance de Enterprise Scrum supera al desarrollo de un producto para el cuál se utiliza Scrum.

[8] http://www.enterprisescrum.com/

1.15 Scaled Agile Framework (SAFe) [9]

El Marco Ágil Escalado (**SAFe**: Scaled Agile Framework) es un conjunto de principios, prácticas y competencias de Lean, Ágil y Desarrollo-Operaciones (DevOps) destinados a guiar a las empresas a escalar prácticas ágiles más allá de un solo equipo.



Este conjunto de fundamentos describe los roles, responsabilidades, artefactos y actividades necesarias para implementar el desarrollo Lean-Agile a escala empresarial sincronizando la alineación, la colaboración y las entregas para un gran número de equipos ágiles comerciales y técnicos.

SAFe combina las metodologías ágiles, el desarrollo de productos Lean y el pensamiento sistémico para ayudar a las empresas a entregar soluciones tecnológicas rápidamente no sólo a nivel de proyecto, sino a nivel de programa, portafolio y de toda la organización.

Valores Fundamentales

1 Alineación: todos los equipos deben apoyar la visión, misión y objetivos estratégicos de la organización.

2 Calidad incorporada: asegurar que cada incremento de la solución refleje estándares de calidad a lo largo del ciclo de vida del desarrollo.

3 Transparencia: compartir información de manera abierta para crear confianza entre los miembros de la organización.

4 Ejecución del programa: los equipos ejecutan y entregan valor continuamente.

[9] © Scaled Agile, Inc., https://www.scaledagileframework.com

Principios

#1 Aplicar una visión económica. Tomar decisiones de desarrollo que consideren los riesgos y los costos de demora, fabricación, operación y desarrollo. Cada desarrollo debe operar dentro de un presupuesto.

#2 Aplicar el pensamiento sistémico. Para mejorar el desarrollo todos deben comprender el objetivo del sistema, ya que optimizar un componente individual no necesariamente mejora el sistema.

#3 Asumir variabilidad y preservar opciones. Mantener múltiples opciones de diseño durante el ciclo de vida del proyecto, ajustando el alcance en base a datos empíricos para optimizar los resultados económicos.

#4 Construir incrementalmente con ciclos de aprendizaje rápidos e integrados. Iteraciones cortas que permitan una respuesta rápida de los clientes sobre el desarrollo para mitigar riesgos de elaborar algo que no valora el cliente.

#5 Basar hitos en la evaluación objetiva de los sistemas de trabajo. Evaluaciones regulares con hitos objetivos para evaluar el producto a lo largo del ciclo de vida del desarrollo.

#6 Visualizar y limitar WIP, reducir los tamaños de lote y administrar longitudes de cola. Limitar el WIP ajustando la demanda a la capacidad real; Reducir el tamaño de los lotes de trabajo para facilitar un flujo rápido; Administrar las longitudes de la cola para reducir los tiempos de espera del nuevo producto.

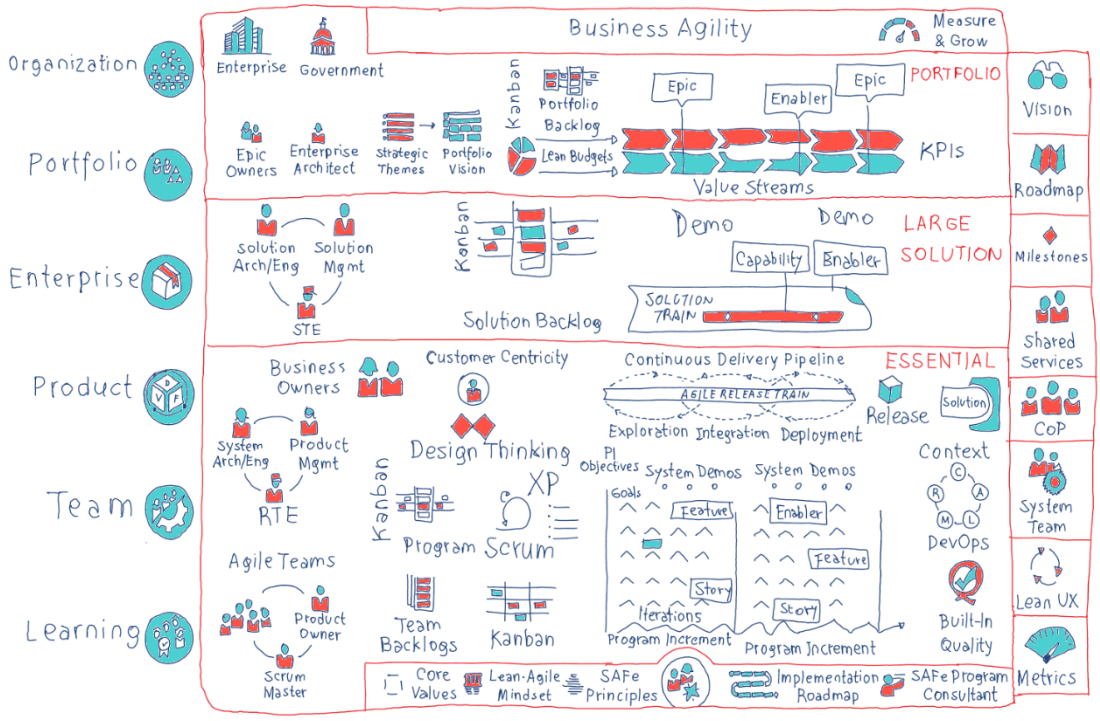
#7 Aplicar cadencia, sincronizar con la planificación entre dominios. El ritmo y sincronización del desarrollo, junto con la planificación periódica entre dominios, permite operar con efectividad en presencia de incertidumbre.

#8 Desbloquear la motivación intrínseca de los trabajadores del conocimiento. Motivar a los trabajadores mediante la autonomía y propósito, minimizando las restricciones, creando un ambiente de influencia mutua.

#9 Descentralizar la toma de decisiones. Empoderar a los trabajadores para tomar decisiones descentralizadas, alineadas con las decisiones estratégicas de la organización, para aumentar la entrega de valor.

#10 Organizar alrededor del valor. La empresa se organiza en torno al valor para entregar más rápidamente. Cuando las demandas del mercado y del cliente cambian, la empresa se reorganiza rápidamente en torno a ese nuevo valor.

El marco SAFe que es de acceso gratuito online.



https://www.scaledagileframework.com/# © Scaled Agile, Inc.

A nivel de equipo, SAFe es muy similar a Scrum para la gestión de un proyecto.

Para programas SAFe utiliza el concepto del Tren de Liberación Ágil (ART) donde varios equipos scrum de 5-10 miembros cada uno, trabajan de manera coordinada para entregar el objetivo de valor comercial del programa.

Para la gestión del portafolio los líderes determinan la visión, objetivos y estrategia de la organización.

1.16 Disciplined Agile (DA) [10]

Disciplined Agile (DA) es un kit de herramientas que simplifica los procesos relacionados con la entrega de soluciones incrementales e iterativas integrando varias prácticas ágiles. A diferencia de otras metodologías que se presentan como un marco metodológico con pasos a seguir, DA ofrece opciones para adaptar los marcos existentes a cada situación en particular, brindando consejos sobre cuándo y cómo aplicar las diferente metodologías.

El kit de herramientas prácticas DA fue diseñado para ofrecer un equilibrio entre los métodos con un enfoque demasiado estrecho (ej. Scrum) o aquellos demasiado detallados (ej. AUP).

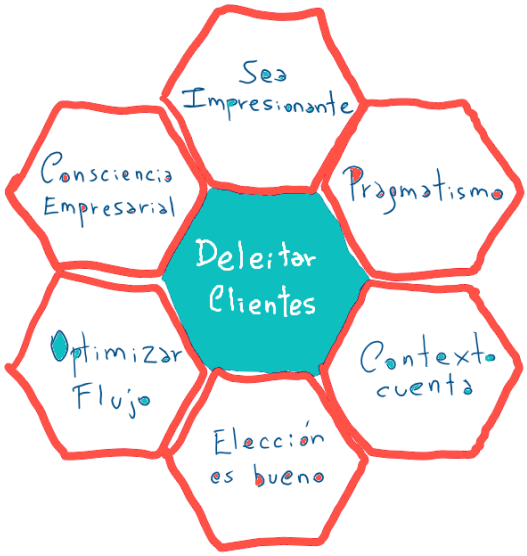
Principios

1. Deleite a los clientes: los productos y servicios no solo satisfacen las necesidades y expectativas del cliente, sino que las superan.

2. Sea impresionante: personas motivadas que reciben el apoyo necesario para cumplir sus objetivos.

3. Pragmatismo: ser lo más efectivo posible.

4. El contexto cuenta: cada persona, cada equipo y cada organización es única, por lo que hay que adaptar la estrategia a cada situación.



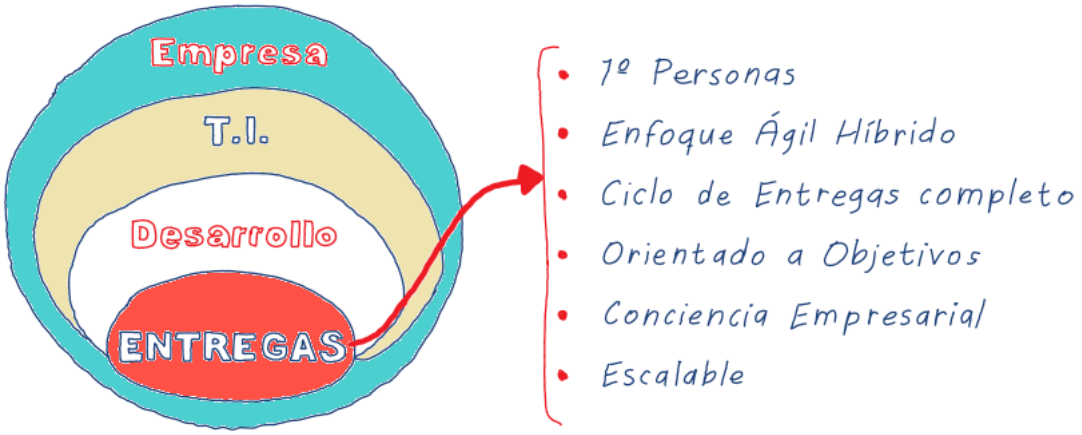
5. La elección es buena: los equipos son dueños de su propio proceso y experimentan opciones para descubrir qué les funciona en la práctica dada la situación a la que se enfrentan.

6. Optimizar el flujo: asegurarse de que los equipos estén bien alineados dentro del sistema adaptativo complejo de su organización.

7. Conciencia empresarial: personas motivadas que consideran las necesidades generales de la organización y no sólo a los objetivos de su equipo.

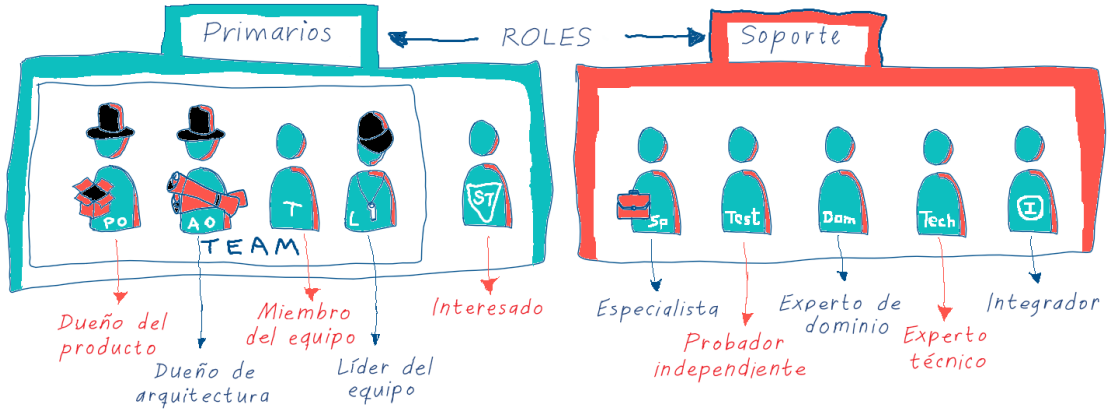
[10] Project Management Institute Inc., https://www.pmi.org/disciplined-agile

El marco DA se divide en cuatro niveles: Entregas (Delivery), Desarrollo y Operaciones (DevOps), Tecnologías de la Información (IT) y Empresa (Enterprise).



Disciplined Agile Delivery es la base de las herramientas DA y tiene las siguientes características: Primero las personas; Enfoque ágil híbrido orientado al aprendizaje; Ciclo de vida de entregas completo; Orientado al objetivo; Consciente de la empresa y Escalable.

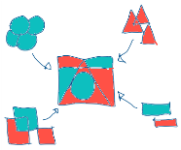
Primero las personas: Roles



Los cinco roles primarios se consideran básicos para cualquier tipo de proyecto. Por su parte, los roles de soporte por lo general aplican a grandes proyectos y suelen ser temporales.

El interesado suele ser algún usuario directo o indirecto, gerente de usuarios, gerente superior, personal de operaciones, patrocinador, auditor, director del programa, etc.

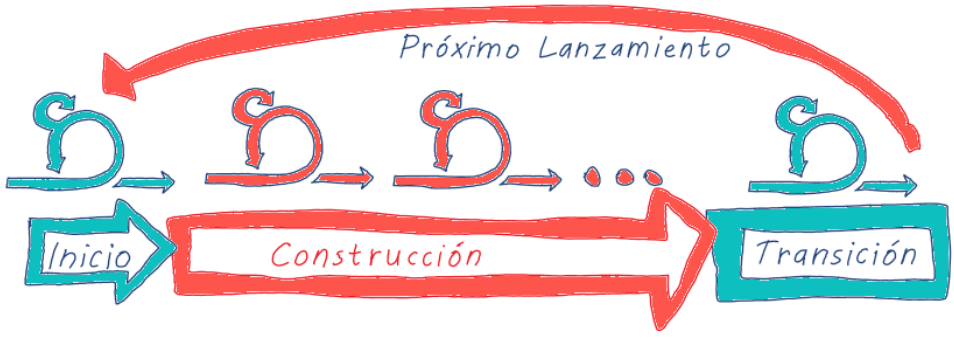
Enfoque ágil híbrido



DA está orientado al aprendizaje de entregas tecnológicas combinando diferentes metodologías tales como: Lean, Kanban, Scrum, XP, FDD, DSDM, AUP, Crystal, LeSS, SAFe, etc.

Ciclo de vida de entregas completo

La metodología admite diferentes ciclo de vida de entregas completos (inicio, construcción, transición) en función del valor y riesgos.



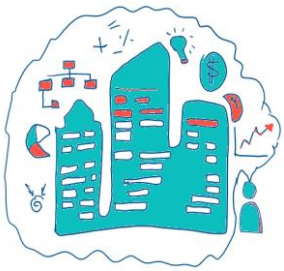
El kit de herramientas DA describe seis ciclos de vida diferentes porque cada equipo se encuentra en una situación única. Por lo tanto, la metodología es lo suficientemente flexible como para admitir varias versiones de un ciclo de vida del proyecto:

1. Ciclo de vida del proyecto Ágil basado en Scrum
2. Ciclo de vida del proyecto Lean basado en Kanban
3. Entrega continua: ciclo de vida Ágil
4. Entrega continua: ciclo de vida Lean
5. Ciclo de vida exploratorio: Lean Startup
6. Ciclo de vida del programa para varios equipos

Orientado al objetivo

DA presenta opciones y flexibilidad para utilizar diferentes metodologías que permitan lograr el objetivo de abordar las necesidades cambiantes de los interesados. DA describe las ventajas y desventajas de cada técnica y en qué situaciones es más adecuada cada una según cuál sea el objetivo.

Consciente de la empresa



Los equipos DA tiene consciencia empresarial, o sea reconocen que forman parte de un ecosistema donde los intereses de la organización están por encima de los intereses del equipo o del proyecto.

Los profesionales en DA trabajan en estrecha colaboración con profesionales de la empresa, como arquitectos empresariales y gerentes de portafolio.

Escalable

Factores de escala para adaptar la estrategia ágil:

2 ← Tamaño del equipo → 1000s
Colocado ← Distribución geográfica → Global
Simple ← Distribución organizacional → Tercerización
Ninguna ← Conformidad → Crítica de vida
Sencillo ← Complejidad de dominio → Muy complejo
Sencillo ← Complejidad técnica → Muy complejo

Ciclo de vida basado en el valor-riesgo
Autoorganización mejorada con un gobierno efectivo
Entrega de soluciones consumibles
Conciencia empresarial sobre conciencia de equipo
Sensible al contexto y orientado a objetivos, no prescriptivo

Ciclo de vida basado en el valor
Equipos autoorganizados
Foco en la construcción de software funcionando
Conciencia de equipo de proyecto
Prescriptivo

1.17 Agilidad organizacional

La **agilidad organizacional** es la capacidad que tiene una compañía de reconocer rápidamente las oportunidades que se presentan en el mercado para adelantarse a la competencia, mejorando su efectividad y productividad en todos los ámbitos y situaciones.

Donald Sull considera que se requieren tres tipos de agilidad en una organización:

- Agilidad estratégica: detectar y cuantificar las oportunidades del mercado con rapidez, flexibilidad e innovación.
- Agilidad de portafolio: gestionar los recursos de manera rápida y eficaz desde todas las áreas del negocio.
- Agilidad operativa: tomar decisiones rápidas y efectivas para aprovechar las oportunidades que la compañía ha detectado.

SAMI (Sidky Agile Measurement Index)

Varios equipos de proyecto se preguntan como comenzar a involucrar mayor agilidad en sus organizaciones y cuál es la mejor metodología para comenzar.

El índice de medición ágil de Sidky (SAMI) es una hoja de ruta de cinco pasos diseñada para guiar a las organizaciones a introducir gradualmente prácticas ágiles en sus procesos.

5 Abarcativo: Establecer una cultura y un ambiente vibrantes para mantener la agilidad.

4 Adaptativo: Responder al cambio a través de múltiples niveles de retroalimentación

3 Integrado: Desarrollar el producto de alta calidad de manera eficiente y efectiva

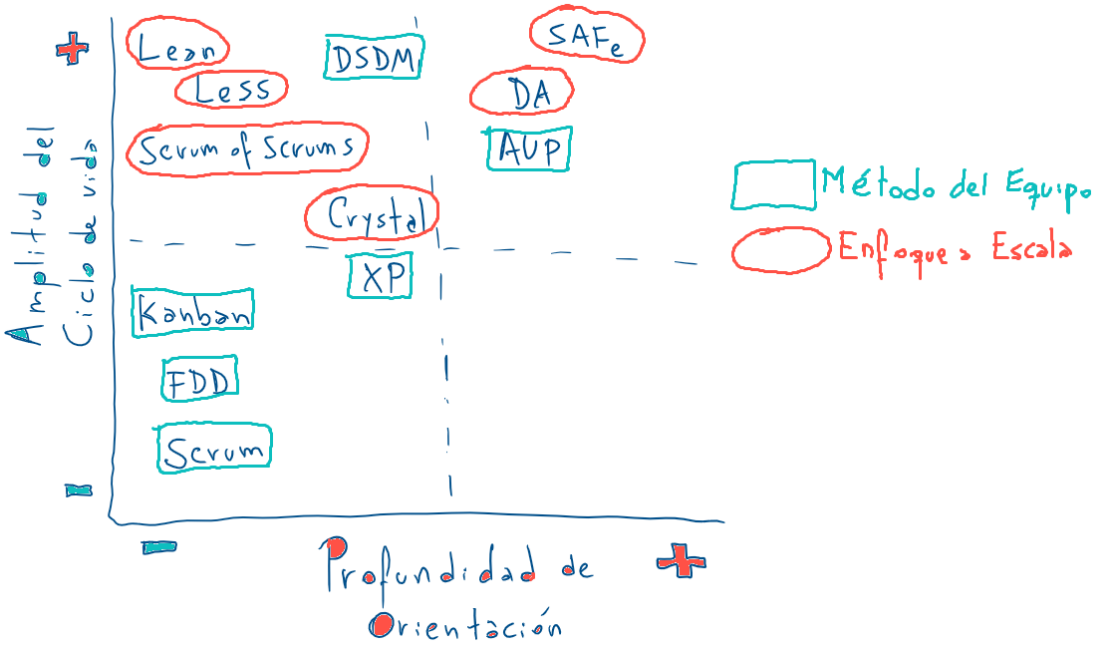
2 Evolutivo: Entrega continua y temprana del producto

1 Colaborativo: Mejorar la comunicación y la colaboración.

Comparación de metodologías ágiles

La agilidad organizacional no tiene una relación directa con las metodologías ágiles que esté utilizando esa organización. Aunque por lo general, aquellas empresas que están familiarizadas con la implementación correcta de metodologías ágiles suelen tener un grado de madurez superior en relación con aquellas organizaciones que utilizan solamente procesos secuenciales con procesos de ciclos de vida predictivos.

En el gráfico a continuación se ubican las diferentes metodologías desarrolladas en este capítulo en función de su profundidad de orientación y su amplitud del ciclo de vida.



Fuente: PMI Agile Practice Guide.

Liderazgo servicial

Independientemente de cuál sea la madurez de agilidad organizacional o la metodología que aplique una organización para agilizar sus proyectos, en todos los casos el perfil del líder de proyecto, o entrenador ágil, o coordinador del equipo es el de un "líder servicial". El líder servicial antepone las necesidades de los miembros del equipo por delante de las propias, concentrando el esfuerzo en el desarrollo y crecimiento del equipo.

Las 10 características fundamentales para el desarrollo de un líder servicial son:

Escucha activa. Demostrar al hablante que el oyente le ha entendido. Escuchar no sólo lo que la persona está expresando directamente, sino también los sentimientos, ideas o pensamientos que subyacen a lo que se está diciendo.

Empatía. Participación afectiva de una persona en los sentimientos de otra persona.

Cura. Ayudar a las personas a resolver sus problemas y conflictos en las relaciones para desarrollar las habilidades de cada individuo.

Conciencia. Capacidad de ver las situaciones desde un enfoque holístico para una mejor comprensión acerca de la ética y los valores.

Persuasión. Habilidad para convencer a una persona mediante razones o argumentos para que piense de una determinada manera o haga cierta cosa para ayudarlo a alcanzar sus objetivos, sin aprovecharse de su poder de coacción.

Conceptualización. Capacidad de ver más allá de los límites de la operación del negocio y centrarse en objetivos a largo plazo.

Prospectiva. Capacidad de prever el resultado probable de una situación. Aprender sobre el pasado y comprender la actualidad para modelar el futuro.

Corresponsabilidad. Obligación de ayudar y servir a los demás para el mayor bienestar de la sociedad.

Compromiso con el crecimiento de las personas. Fomentar e invertir en el crecimiento personal, profesional y espiritual de los miembros del equipo.

Construcción de la comunidad. Construir una comunidad fuerte dentro de la organización y desarrollar una comunidad con otras empresas e instituciones.

1.18 Ejercicios – Mentalidad Ágil

1 Coloque el ciclo de vida del proyecto (Predictivo, Incremental, Iterativo, Ágil) de cada uno de los siguientes enunciados.

	Ciclo de vida
Combina ciclos iterativos e incrementales, realizando iteraciones sobre un producto para obtener entregables intermedios listos para usar en cada lanzamiento a lo largo del ciclo del proyecto.	
En las primeras iteraciones se entrega una funcionalidad básica y se va agregando mayor funcionalidad al producto a medida que avanzan las fases del proyecto	
En las primeras iteraciones se va construyendo un borrador del producto final mediante el análisis-desarrollo-reflexión y en las fases sucesivas se va agregando calidad al producto con más análisis-desarrollo-reflexión.	
Hasta que no finaliza o está avanzada la fase predecesora, no comienza su sucesora	

2 Coloque en cada uno de los siguientes cuadrantes el ciclo de vida del proyecto (Predictivo, Incremental, Iterativo, Ágil) más recomendado.

Frecuencia de entregas		
Alta		
Baja		
	Bajo — Cambios del mercado — Alto	

3 Complete los cuadros en blanco de la siguiente tabla con las palabras “fijo” o “flexible”.

Ciclo de vida	Costo	Tiempo	Alcance
Predictivo			
Adaptativo			

4 Unir con flechas los siguientes conceptos del Manifiesto Ágil

Colaboración con el cliente	sobre documentación extensiva
Individuos e interacciones	sobre negociación contractual
Respuesta ante el cambio	sobre procesos y herramientas
Software funcionando	sobre seguir un plan

5 Marque con una cruz a qué filosofía corresponde cada uno de los siguientes principios.

Principio	Manifiesto Ágil	Declaración de la Interdependencia
Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.		
Disparamos el rendimiento mediante la responsabilidad común sobre los resultados y sobre la propia efectividad del equipo.		
Entregamos software funcional frecuentemente, entre dos semanas y dos meses.		
Esperamos lo inesperado y lo gestionamos mediante iteraciones, anticipación y adaptación.		
Incrementamos el retorno de la inversión centrándonos en un flujo continuo de valor.		
La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.		
Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua con valor.		

6 Coloque del 1 a 5 según el orden cronológico del Marco Ágil propuesto por Jim Highsmith.

#	Fases
	Adaptación (Adapt)
	Cierre (Close)
	Especulación (Speculate)
	Exploración (Explore)
	Visualización (Envision)

7 Complete en qué fase del Marco Ágil de Jim Highsmith se realiza cada una de las siguientes actividades.

Fase	Actividad
	Aprender e incorporar el aprendizaje al siguiente equipo de proyecto
	Crear una declaración del alcance del producto y sus criterios de éxito
	Desarrollar un plan de lanzamientos, hitos e iteraciones
	Entregar funcionalidades en un corto período de tiempo
	Realizar correcciones, incorporar y retener las lecciones aprendidas

8 ¿Cuáles son las 6 capas de la cebolla de planificación ágil propuestas por Mike Cohn?

#	Capa
1	
2	
3	
4	
5	
6	

9 Complete las palabras que faltan en cada uno de los principios del pensamiento Lean.

1. Especificar con precisión el _____ de cada proyecto
2. Definir el _____ de valor del proyecto
3. Permitir que el valor _____ sin interrupciones
4. Permitir que el _____ participe en la identificación de "valor"
5. Buscar de manera _____ la perfección

10 Nombre 3 de los 7 desechos más comunes que hay que eliminar del flujo de valor.

11 En base a la filosofía Lean, complete el párrafo con alguna de las siguientes frases:
Lo antes posible / Lo más tarde posible / En el último momento responsable

Para no entorpecer el flujo de valor con demoras, deberíamos decidir _____ buscando el punto óptimo de la compensación entre el tiempo disponible para una decisión y la necesidad de desarrollar un producto.

12 Marque con una cruz cuál es un principio y cuál una práctica de Kanban.

	Principio	Práctica
Comenzar aplicando el método en los procesos actuales y luego estimular cambios.		
Limitar el trabajo en progreso (WIP) para estimular los cambios.		
Perseguir el cambio incremental, gradual y evolutivo.		
Reconocer oportunidades de mejora mediante cambios continuos, graduales y evolutivos.		
Respetar el proceso actual, los roles las responsabilidades y los cargos.		
Visualizar el flujo de trabajo en columnas y hacerlo visible para comprender cómo avanza el trabajo.		

13 ¿Con qué frecuencia (trimestral, mensual, quincenal, semanal, diaria) se realiza cada uno de los siguientes bucles de retroalimentación en el sistema Kanban?

Frecuencia	Bucle de retroalimentación
	Reunión de reabastecimiento: identificar los elementos en los que trabajará el equipo a continuación.
	Reunión diaria: El equipo coordina sus actividades para el día.
	Revisión de entrega de servicio: examinar y mejorar la efectividad de un servicio.
	Revisión de estrategia: definir servicios y el contexto.
	Revisión de operaciones: evaluar el equilibrio entre servicios, personas y recursos para maximizar la entrega de valor.
	Revisión de riesgos: evaluar y responder a los riesgos de entrega en los servicios.

14 Marque con una cruz si los siguientes ítems son Artefactos o Eventos de Scrum

	Artefacto	Evento
Incremento del producto (Increment)		
Planificación de la iteración (Sprint planning)		
Retrospectiva de la iteración (Sprint retrospective)		
Reunión diaria (Daily Scrum)		
Revisión de la iteración (Sprint review)		
Trabajo pendiente asociado al producto (Product backlog)		
Trabajo pendiente de la iteración (Sprint backlog)		

15 Coloque el orden cronológico (1 a 7) de los siguientes ítems de Scrum

#	
	Incremento del producto (Increment)
	Planificación de la iteración (Sprint planning)
	Retrospectiva de la iteración (Sprint retrospective)
	Reunión diaria (Daily Scrum)
	Revisión de la iteración (Sprint review)
	Trabajo pendiente asociado al producto (Product backlog)
	Trabajo pendiente de la iteración (Sprint backlog)

16 Coloque la duración de cada una de las siguientes reuniones para un proyecto que aplica Scrum con iteraciones de 2 semanas.

Reunión	_____ Duración _____
Diaria	
Revisión de la iteración	
Retrospectiva de la iteración	

17 Marque quién cumple los siguientes roles en un equipo Scrum.

Rol	Dueño del producto	Equipo de desarrollo	Scrum Master
Acepta o rechaza el incremento del producto			
Aísla al equipo de interferencias externas			
Asegura que el product backlog sea visible			
Decide cómo desarrollar el producto			
Entrena a la organización en la adopción de Scrum			
Prioriza los componentes del product backlog			

18 Según ScrumBan, marque verdadero (V) o falso (F).

	V	F
El desencadenante de planificación está asociado con la cantidad de tareas que quedan en la sección "Tareas pendientes" del tablero.		
Las reuniones de planificación se llevan a cabo sólo cuando sea necesario determinar qué historias de usuarios se completarán en la próxima iteración.		
Los roles que tiene un equipo antes de adoptar Scrumban se mantienen.		
Para mantener las iteraciones cortas, se mantiene alto el límite de trabajo en progreso (WIP).		
Requiere un número máximo de miembros del equipo.		

19 Una con flechas las siguientes frases con los valores de XP.

Comunicación	Dejar de hacer lo que no funciona
Coraje	Evitar desperdicios
Respeto	Honrar la relación del equipo
Retroalimentación	Identificar áreas de mejora
Sencillez	Transferir el conocimiento

20 Según las siguientes prácticas de XP, complete verdadero o falso.

	V	F
Cada iteración es de 2 semanas		
Cada persona debe trabajar un máximo de 40 horas semanales		
Construir el sistema y ejecutar todas las pruebas en diez minutos		
El software lo desarrollan dos personas sentadas en la misma máquina		
La refactorización forma parte del diseño incremental		
Se suelen agregar historias de baja prioridad en los ciclos semanales		
Seguir el ciclo “desarrollar código -> escribir pruebas -> ejecutar pruebas”		
Utilizar radiadores de información para comunicar actualizaciones.		

21 Identifique los principales roles de XP.

Rol de XP	SI	NO
Cliente		
Desarrollador		
Dueño de arquitectura		
El entrenador		
Experto de dominio		
Integrador		
Patrocinador		
Rastreador		

22 Ordene cronológicamente del 1 a 5 los procesos de Feature Driven Development (FDD)

#	Proceso
	Construir por funcionalidad
	Desarrollar lista de funcionalidades
	Desarrollar modelo general
	Diseñar por funcionalidad
	Planificar por funcionalidad

23 Coloque las siguientes prácticas de FDD en la tabla a continuación: Modelado de objetos de dominio; Desarrollando por funcionalidad; Propiedad de código individual; Equipos por funcionalidad; Inspecciones; Gestión de la configuración; Construcciones regulares; Visibilidad de progreso y resultados.

Práctica	
	Cualquier función demasiado compleja para implementarse en dos semanas se descompone en funciones más pequeñas.
	Demostraciones frecuentes del producto al cliente para detectar errores de integración de manera temprana.
	Detección de defectos para garantizar un diseño de buena calidad.
	Equipos pequeños desarrollan cada funcionalidad.
	Explorar y explicar el dominio del problema a resolver
	Identificar el código fuente de todas las características que van completando y a mantener un historial de cambios.
	Informes de avance frecuentes, apropiados y precisos
	Se asignan distintas piezas de códigos a un solo propietario responsable de la coherencia, el rendimiento y la integridad.

24 Ordene cronológicamente (1 a 7) las fases del proyecto según Dynamic System Development Method (DSDM).

#	Fase
	Caso de negocios
	Diseñar y construir iteraciones
	Estudio de factibilidad
	Fase posterior al proyecto
	Implementación
	Iteración del modelo funcional
	Pre Proyecto

25 Complete el significado de MoSCoW para la priorización de requisitos:

Inglés		Español
M __ __ __	have	
S __ __ __ __ __	have	
C __ __ __ __	have	
W __ __ __ __	like to have	

26 Marque la tabla con verdadero o falso según DSDM.

	V	F
Apoyo entre el cliente y el proveedor		
Creación de prototipos funcionales		
El equipo escribe propuestas formales a la alta gerencia		
Los cambios son reversibles		
Pruebas integradas durante todo el ciclo de vida		
Retroalimentación de los usuarios sobre los requisitos		

27 Complete con alguna de las siguientes palabras: “serial”, “iterativo” o “incremental”.

El Agile Unified Process (AUP) es un proceso de desarrollo de software iterativo e _____; “_____ en lo grande” e “_____ en lo pequeño”.

28 Complete la tabla con las fases de AUP (Inicio, Elaboración, Construcción, Transición)

Fase	
	Ampliar la comprensión de los requisitos del producto.
	Completar el desarrollo del producto.
	Completar las pruebas del producto para su lanzamiento.
	Definir la arquitectura del producto.

29 Complete la tabla con las disciplinas de AUP: Modelo, Implementación, Prueba, Despliegue, Gestión de la configuración, Gestión del proyecto, Ambiente

Disciplina	
	Asegurarse que el proceso y las herramientas estén disponibles.
	Comprender el negocio de la organización y el dominio del problema
	Gestionar las actividades del proyecto, personas y entregas.
	Gestionar los cambios.
	Planificar la entrega del producto y ejecutar el plan.
	Realizar una evaluación objetiva para garantizar la calidad.
	Transformar los modelos en código ejecutable.

30 Según Crystal, complete con las siguientes propiedades en la tabla: Entregas frecuentes; Mejora reflexiva; Comunicación osmótica; Seguridad personal; Enfoque; Fácil acceso a usuarios expertos; Pruebas automatizadas.

Propiedad	Descripción
	Acceso mínimo semanalmente
	Ciclo de vida iterativo e incremental
	Equipo en el mismo lugar físico
	Gestión de la configuración e integración
	Mejora continua en cada iteración
	Todos pueden expresar su opinión
	Trabajo sin interrupciones durante 2 horas

31 Unir con flechas según la metodología Crystal.

Metodología Crystal	Tamaño del equipo
Amarillo	1-6 personas
Claro	7-20 personas
Marrón	21-40 personas
Naranja	41-80 personas
Rojo	81-200 personas

32 Marque la tabla con verdadero o falso según Scrum of Scrums.

	V	F
Es necesario asignar un coordinar para SoS		
La reunión SoS se recomienda que sea diaria		
La reunión SoS suele demorar 2 horas		
Los dueños del producto de cada equipo participan de SoS		
Los Scrum Masters son los únicos que participan en SoS		
Todos los miembros de cada equipo Scrum participan de SoS		

33 Indique cuál de los siguientes ítems se mantiene según Scrum y cuáles serían nuevos agregados de LeSS.

	Se mantiene	Se agrega
A mitad de la iteración se refina el product backlog		
Dueño del producto único para todos		
Equipos multifuncionales completos en cada equipo		
Incremento del mismo producto al final de cada iteración		
La planificación del Sprint se divide en dos partes		
Los equipos comparten información de lo que están realizando		
Product backlog único para todos		
Retrospectiva global de la iteración		
Varios equipos Scrum trabajan sobre el mismo producto		

34 Marque los 3 principales marcos en los que se basa SAFe

Marco	Utiliza SAFe
Ágil	
AUP	
DA	
DevOps	
FDD	
Kanban	
Lean	
Scrum	
XP	

35 Marque los 4 valores fundamentales de SAFe.

Valor fundamental	SAFe
Alineación	
Calidad incorporada	
Comunicación	
Coraje	
Ejecución del programa	
Respeto	
Retroalimentación	
Sencillez	
Transparencia	

36 Coloque en la tabla los siguientes principios de SAFe: Visión económica; Asumir variabilidad; Visualizar y limitar WIP; Desbloquear la motivación intrínseca; Descentralizar la toma de decisiones; Organizar alrededor del valor.

Principio	Descripción
	Ajustar la demanda a la capacidad real.
	Cuando las demandas del mercado y del cliente cambian, la empresa se reorganiza rápidamente.
	Empoderar a los trabajadores.
	Mantener múltiples opciones de diseño durante el ciclo de vida del proyecto.
	Otorgar a los trabajadores autonomía y propósito.
	Tomar decisiones de desarrollo que consideren los riesgos y los costos de demora, fabricación, operación y desarrollo.

37 Coloque en la tabla los siguientes principios de Disciplined Agile (DA): Deleite a los clientes; Sea impresionante; Pragmatismo; El contexto cuenta; La elección es buena; Optimizar el flujo; Conciencia empresarial.

Principio	Descripción
	Los productos superan las necesidades y expectativas del cliente
	Personas motivadas que reciben apoyo
	Ser lo más efectivo posible
	Cada persona, cada equipo y cada organización es única
	Los equipos experimentan opciones
	Equipos alineados dentro del sistema adaptativo complejo
	Considerar las necesidades generales de la organización

38 Marque los roles primarios y de soporte de Disciplined Agile (DA).

Rol	Primario	Soporte
Dueño de arquitectura		
Dueño del producto		
Especialista		
Experto de Dominio		
Experto Técnico		
Integrador		
Interesado		
Líder del equipo		
Miembro del equipo		
Probador independiente		

39 Marque en la tabla cuáles enfoques utiliza Disciplined Agile

Metodología	
AUP	
Crystal	
DSDM	
FDD	
Kanban	
Lean	
LeSS	
SAFe	
Scrum	
XP	

40 Completar en la tabla los distintos tipos de agilidad organizacional: estratégica, portafolio, operativa.

Agilidad	Concepto
	Detectar y cuantificar las oportunidades del mercado con rapidez, flexibilidad e innovación
	Tomar decisiones rápidas y efectivas para aprovechar las oportunidades de la compañía
	Gestionar los recursos de manera rápida y eficaz desde todas las áreas del negocio

41 Coloque en cada uno de los cuadrantes las siguientes metodologías: Lean, Kanban, Scrum, XP, FDD, DSDM, AUP, Crystal, Scrum of Scrums, LeSS, SAFe, DA.

Amplitud del ciclo de vida: Alto / Bajo		
	Baja — Profundidad de Orientación — Alta	

42 Marque cuáles de las siguientes características aplican a un Líder Servicial.

Metodología	
Compromiso con el crecimiento de las personas	
Conceptualización	
Conciencia	
Construcción de la comunidad	
Corresponsabilidad	
Cura	
Empatía	
Escucha activa	
Persuasión	
Prospectiva	

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

43 En su organización están trabajando con dos proyectos en paralelo y dos equipos diferentes. Para el proyecto A se aplican técnicas Scrum y para B se aplica Lean Kanban. ¿Cuál será el tamaño de cada equipo?

A. Proyecto A 10 personas y Proyecto B 7 personas
B. Proyecto A 8 personas y Proyecto B 2 personas
C. Proyecto A 2 personas y Proyecto B 8 personas
D. Proyecto A 7 personas y Proyecto B 15 personas

44 En un proyecto sobre robótica, se van a implementar metodologías ágiles. Algunos ingenieros acostumbrados a trabajar con metodologías de ciclo predictivo están preocupados por la corrupción del alcance que tuvieron en otros proyectos. ¿Qué explicará usted a esos ingenieros?

A. Durante una iteración, no agregaremos nuevas tareas
B. No agregaremos alcance adicional al proyecto para evitar que ocurra lo mismo del pasado
C. La corrupción del alcance no existe en proyectos que aplican metodologías ágiles
D. Los desarrolladores priorizarán los nuevos trabajos para mitigar la corrupción del alcance

45 Durante el desarrollo de un producto para que vuelen personas de manera autónoma, se están utilizando metodologías ágiles. Uno de los interesados claves de este proyecto le pide explicaciones al scrum master de porqué han eliminado de esta iteración una de las funcionalidades que agrega muchísimo valor y tiene bajo costo. ¿Qué debería responder el scrum master a ese interesado?

A. Analizaremos la situación con los miembros del equipo y le daremos una respuesta lo antes posible
B. Solicite una reunión con el dueño del producto para que atienda su consulta
C. El analista de negocios podrá dar respuesta a su consulta
D. En esta iteración no tuvimos tiempo de incorporarlo, pero será la primera historia de usuario a considerar en la próxima iteración

1.19 Respuesta ejercicios – Mentalidad Ágil

Ciclo de vida del proyecto

	Ciclo de vida
Combina ciclos iterativos e incrementales, realizando iteraciones sobre un producto para obtener entregables intermedios listos para usar en cada lanzamiento a lo largo del ciclo del proyecto.	Ágil
En las primeras iteraciones se entrega una funcionalidad básica y se va agregando mayor funcionalidad al producto a medida que avanzan las fases del proyecto	Incremental
En las primeras iteraciones se va construyendo un borrador del producto final mediante el análisis-desarrollo-reflexión y en las fases sucesivas se va agregando calidad al producto con más análisis-desarrollo-reflexión.	Iterativo
Hasta que no finaliza o está avanzada la fase predecesora, no comienza su sucesora	Predictivo

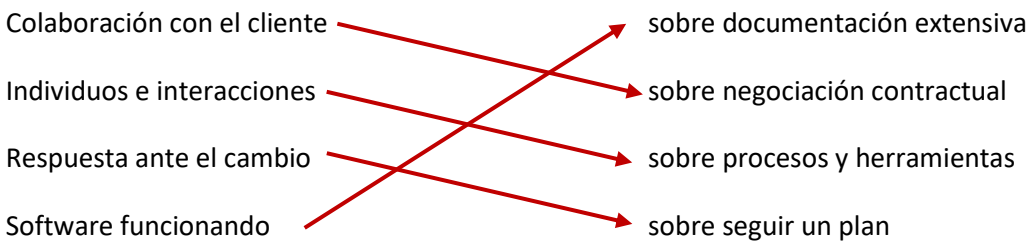
Ciclo de vida del proyecto

Frecuencia de entregas		
Alta	Incremental	Ágil
Baja	Predictivo	Iterativo
	Bajo — Cambios del mercado — Alto	

3R Ciclos de vida

Ciclo de vida	Costo	Tiempo	Alcance
Predictivo	flexible	flexible	fijo
Adaptativo	fijo	fijo	flexible

4R Manifiesto Ágil



5R Manifiesto Ágil y Declaración de la Interdependencia

Principio	Manifiesto Ágil	Declaración de la Interdependencia
Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.	√	
Disparamos el rendimiento mediante la responsabilidad común sobre los resultados y sobre la propia efectividad del equipo.		√
Entregamos software funcional frecuentemente, entre dos semanas y dos meses.	√	
Esperamos lo inesperado y lo gestionamos mediante iteraciones, anticipación y adaptación.		√
Incrementamos el retorno de la inversión centrándonos en un flujo continuo de valor.		√
La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.	√	
Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua con valor.	√	

6R Marco Ágil

#	Fases
4º	Adaptación (Adapt)
5º	Cierre (Close)
2º	Especulación (Speculate)
3º	Exploración (Explore)
1º	Visualización (Envision)

7R Marco Ágil

Fase	Actividad
Cierre	Aprender e incorporar el aprendizaje al siguiente equipo de proyecto
Visualización	Crear una declaración del alcance del producto y sus criterios de éxito
Especulación	Desarrollar un plan de lanzamientos, hitos e iteraciones
Exploración	Entregar funcionalidades en un corto período de tiempo
Adaptación	Realizar correcciones, incorporar y retener las lecciones aprendidas

8R Cebolla de planificación ágil

#	Capa
1	Estrategia
2	Portafolio
3	Producto
4	Lanzamiento
5	Iteración
6	Día

9R Pensamiento Lean.

Especificar con precisión el valor de cada proyecto

Definir el flujo de valor del proyecto

Permitir que el valor fluya sin interrupciones

Permitir que el cliente participe en la identificación de "valor"

Buscar de manera continua la perfección

10R Desechos a eliminar del flujo de valor

Transporte	Inventarios	Defectos

Los "7 desechos" más comunes que hay que eliminar del flujo de valor son: transporte, inventarios, movimiento, esperas, sobre procesamiento, sobreproducción y defectos.

11R Filosofía Lean

Para no entorpecer el flujo de valor con demoras, deberíamos decidir en el último momento responsable buscando el punto óptimo de la compensación entre el tiempo disponible para una decisión y la necesidad de desarrollar un producto.

12R Kanban.

	Principio	Práctica
Comenzar aplicando el método en los procesos actuales y luego estimular cambios.	√	
Limitar el trabajo en progreso (WIP) para estimular los cambios.		√
Perseguir el cambio incremental, gradual y evolutivo.	√	
Reconocer oportunidades de mejora mediante cambios continuos, graduales y evolutivos.		√
Respetar el proceso actual, los roles las responsabilidades y los cargos.	√	
Visualizar el flujo de trabajo en columnas y hacerlo visible para comprender cómo avanza el trabajo.		√

13R Kanban

Frecuencia	Bucle de retroalimentación
semanal	Reunión de reabastecimiento: identificar los elementos en los que trabajará el equipo a continuación.
diaria	Reunión diaria: El equipo coordina sus actividades para el día.
quincenal	Revisión de entrega de servicio: examinar y mejorar la efectividad de un servicio.
trimestral	Revisión de estrategia: definir servicios y el contexto.
mensual	Revisión de operaciones: evaluar el equilibrio entre servicios, personas y recursos para maximizar la entrega de valor.
mensual	Revisión de riesgos: evaluar y responder a los riesgos de entrega en los servicios.

14R Scrum

	Artefacto	Evento
Incremento del producto (Increment)	√	
Planificación de la iteración (Sprint planning)		√
Retrospectiva de la iteración (Sprint retrospective)		√
Reunión diaria (Daily Scrum)		√
Revisión de la iteración (Sprint review)		√
Trabajo pendiente asociado al producto (Product backlog)	√	
Trabajo pendiente de la iteración (Sprint backlog)	√	

15R Scrum

#	
6º	Incremento del producto (Increment)
2º	Planificación de la iteración (Sprint planning)
7º	Retrospectiva de la iteración (Sprint retrospective)
4º	Reunión diaria (Daily Scrum)
5º	Revisión de la iteración (Sprint review)
1º	Trabajo pendiente asociado al producto (Product backlog)
3º	Trabajo pendiente de la iteración (Sprint backlog)

16R Scrum

Reunión	Duración
Diaria	15 minutos
Revisión de la iteración	2 horas (1 hora por semana de iteración)
Retrospectiva de la iteración	1,5 horas (45 minutos por semana de iteración)

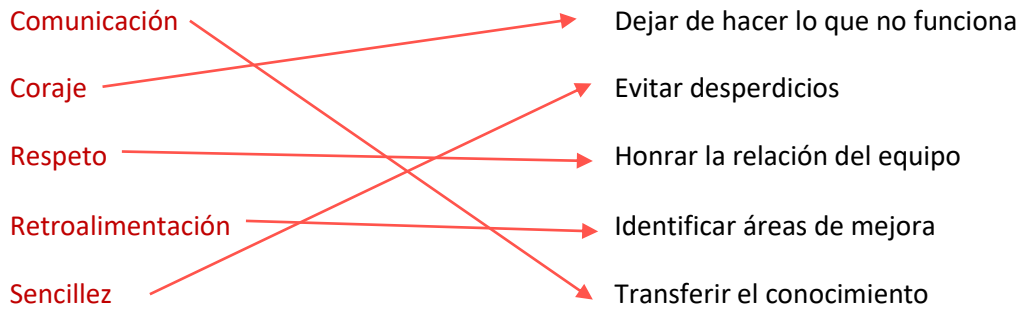
17R Scrum

Rol	Dueño del producto	Equipo de desarrollo	Scrum Master
Acepta o rechaza el incremento del producto	√		
Aísla al equipo de interferencias externas			√
Asegura que el product backlog sea visible	√		
Decide cómo desarrollar el producto		√	
Entrena a la organización en la adopción de Scrum			√
Prioriza los componentes del product backlog	√		

18R ScrumBan

	V	F
El desencadenante de planificación está asociado con la cantidad de tareas que quedan en la sección "Tareas pendientes" del tablero.	√	
Las reuniones de planificación se llevan a cabo sólo cuando sea necesario determinar qué historias de usuarios se completarán en la próxima iteración.	√	
Los roles que tiene un equipo antes de adoptar Scrumban se mantienen.	√	
Para mantener las iteraciones cortas, se mantiene alto el límite de trabajo en progreso (WIP).		√
Requiere un número máximo de miembros del equipo.		√

19R XP



Comunicación: transferir el conocimiento entre los miembros del equipo mediante discusiones cara a cara y dibujos. / Sencillez: evitar desperdicios realizando sólo las cosas absolutamente necesarias. / Retroalimentación: recibir comentarios constantes sobre el trabajo para identificar áreas de mejora y revisar las prácticas. / Coraje: plantear problemas que reducen la eficacia de su equipo; dejar de hacer lo que no funciona y probar otra cosa. / Respeto: compartir comentarios que honren la relación del equipo, trabajando juntos para identificar soluciones simples.

20R XP

	V	F
Cada iteración es de 2 semanas		√
Cada persona debe trabajar un máximo de 40 horas semanales	√	
Construir el sistema y ejecutar todas las pruebas en diez minutos	√	
El software lo desarrollan dos personas sentadas en la misma máquina	√	
La refactorización forma parte del diseño incremental	√	
Se suelen agregar historias de baja prioridad en los ciclos semanales	√	
Seguir el ciclo “desarrollar código -> escribir pruebas -> ejecutar pruebas”		√
Utilizar radiadores de información para comunicar actualizaciones.	√	

21R XP

Rol de XP	SI	NO
Cliente	√	
Desarrollador	√	
Dueño de arquitectura		√
El entrenador	√	
Experto de dominio		√
Integrador		√
Patrocinador		√
Rastreador	√	

22R Feature Driven Development (FDD)

#	Proceso
5º	Construir por funcionalidad
2º	Desarrollar lista de funcionalidades
1º	Desarrollar modelo general
4º	Diseñar por funcionalidad
3º	Planificar por funcionalidad

23R FDD

Desarrollando por funcionalidad	Cualquier función demasiado compleja para implementarse en dos semanas se descompone en funciones más pequeñas.
Construcciones regulares	Demostraciones frecuentes del producto al cliente para detectar errores de integración de manera temprana.
Inspecciones	Detección de defectos para garantizar un diseño de buena calidad.
Equipos por funcionalidad	Equipos pequeños desarrollan cada funcionalidad.
Modelado de objetos de dominio	Explorar y explicar el dominio del problema a resolver
Gestión de la configuración	Identificar el código fuente de todas las características que van completando y a mantener un historial de cambios.
Visibilidad de progreso y resultados	Informes de avance frecuentes, apropiados y precisos
Propiedad de código individual	Se asignan distintas piezas de códigos a un solo propietario responsable de la coherencia, el rendimiento y la integridad.

24R Dynamic System Development Method (DSDM)

#	Fase
3º	Caso de negocios
5º	Diseñar y construir iteraciones
2º	Estudio de factibilidad
7º	Fase posterior al proyecto
6º	Implementación
4º	Iteración del modelo funcional
1º	Pre Proyecto

25R DSDM – MoSCoW

Inglés		Español
Must	have	DEBE tener este requisito de manera obligatoria
Should	have	DEBERÍA tener este requisito si fuera posible
Could	have	PODRÍA tener este requisito si se puede entregar
Would	like to have	ME GUSTARÍA agregar estos requisitos si queda tiempo

26R DSDM

	V	F
Apoyo entre el cliente y el proveedor	√	
Creación de prototipos funcionales	√	
El equipo escribe propuestas formales a la alta gerencia		√
Los cambios son reversibles	√	
Pruebas integradas durante todo el ciclo de vida	√	
Retroalimentación de los usuarios sobre los requisitos	√	

27R AUP

El Agile Unified Process (AUP) es un proceso de desarrollo de software iterativo e incremental; "serial en lo grande" e "iterativo en lo pequeño".

28R AUP

Fase	
Elaboración	Ampliar la comprensión de los requisitos del producto.
Construcción	Completar el desarrollo del producto.
Transición	Completar las pruebas del producto para su lanzamiento.
Inicio	Definir la arquitectura del producto.

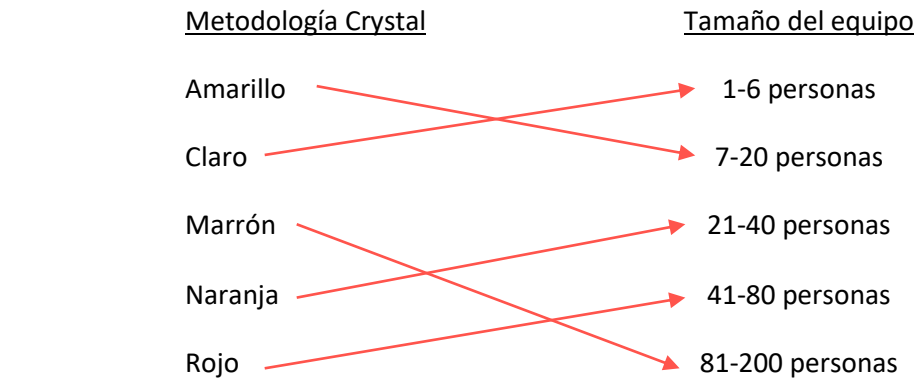
29R AUP

Disciplina	
Ambiente	Asegurarse que el proceso y las herramientas estén disponibles.
Modelo	Comprender el negocio de la organización y el dominio del problema
Gestión proyecto	Gestionar las actividades del proyecto, personas y entregas.
Gestión configuración	Gestionar los cambios.
Despliegue	Planificar la entrega del producto y ejecutar el plan.
Prueba	Realizar una evaluación objetiva para garantizar la calidad.
Implementación	Transformar los modelos en código ejecutable.

30R Crystal

Propiedad	Descripción
Fácil acceso a usuarios expertos	Acceso mínimo semanalmente
Entregas frecuentes	Ciclo de vida iterativo e incremental
Comunicación osmótica	Equipo en el mismo lugar físico
Pruebas automatizadas	Gestión de la configuración e integración
Mejora reflexiva	Mejora continua en cada iteración
Seguridad personal	Todos pueden expresar su opinión
Enfoque	Trabajo sin interrupciones durante 2 horas

31R Crystal



32R Scrum of Scrums

	V	F
Es necesario asignar un coordinar para SoS		√
La reunión SoS se recomienda que sea diaria		√
La reunión SoS suele demorar 2 horas		√
Los dueños del producto de cada equipo participan de SoS		√
Los Scrum Masters son los únicos que participan en SoS		√
Todos los miembros de cada equipo Scrum participan de SoS		√

33R LeSS

	Se mantiene	Se agrega
A mitad de la iteración se refina el product backlog		√
Dueño del producto único para todos	√	
Equipos multifuncionales completos en cada equipo	√	
Incremento del mismo producto al final de cada iteración	√	
La planificación del Sprint se divide en dos partes		√
Los equipos comparten información de lo que están realizando		√
Product backlog único para todos	√	
Retrospectiva global de la iteración		√
Varios equipos Scrum trabajan sobre el mismo producto		√

34R SAFe

Marco	Utiliza SAFe
Ágil	√
AUP	
DA	
DevOps	√
FDD	
Kanban	
Lean	√
Scrum	
XP	

35R SAFe

Valor fundamental	SAFe	XP
Alineación	√	
Calidad incorporada	√	
Comunicación		√
Coraje		√
Ejecución del programa	√	
Respeto		√
Retroalimentación		√
Sencillez		√
Transparencia	√	

36R SAFe

Principio	Descripción
Visualizar y limitar WIP	Ajustar la demanda a la capacidad real.
Organizar alrededor del valor	Cuando las demandas del mercado y del cliente cambian, la empresa se reorganiza rápidamente.
Descentralizar la toma de decisiones	Empoderar a los trabajadores.
Asumir variabilidad	Mantener múltiples opciones de diseño durante el ciclo de vida del proyecto.
Desbloquear la motivación intrínseca	Otorgar a los trabajadores autonomía y propósito.
Visión económica	Tomar decisiones de desarrollo que consideren los riesgos y los costos de demora, fabricación, operación y desarrollo.

37R Disciplined Agile (DA)

Principio	Descripción
Deleite a los clientes	Los productos superan las necesidades y expectativas del cliente
Sea impresionante	Personas motivadas que reciben apoyo
Pragmatismo	Ser lo más efectivo posible
El contexto cuenta	Cada persona, cada equipo y cada organización es única
La elección es buena	Los equipos experimentan opciones
Optimizar el flujo	Equipos alineados dentro del sistema adaptativo complejo
Conciencia empresarial	Considerar las necesidades generales de la organización

38R DA

Rol	Primario	Soporte
Dueño de arquitectura	√	
Dueño del producto	√	
Especialista		√
Experto de Dominio		√
Experto Técnico		√
Integrador		√
Interesado	√	
Líder del equipo	√	
Miembro del equipo	√	
Probador independiente		√

39R DA

Metodología	
AUP	√
Crystal	√
DSDM	√
FDD	√
Kanban	√
Lean	√
LeSS	√
SAFe	√
Scrum	√
XP	√

40R Agilidad organizacional

Agilidad	Concepto
Estratégica	Detectar y cuantificar las oportunidades del mercado con rapidez, flexibilidad e innovación
Operativa	Tomar decisiones rápidas y efectivas para aprovechar las oportunidades de la compañía
Portafolio	Gestionar los recursos de manera rápida y eficaz desde todas las áreas del negocio

41R Metodologías Ágiles

Amplitud del ciclo de vida		
Alto	Lean LeSS DSDM Scrum of Scrums Crystal	SAFe DA AUP
Bajo	XP Kanban FDD Scrum	
	Baja — Profundidad de Orientación — Alta	

42R Líder Servicial

Metodología	
Compromiso con el crecimiento de las personas	√
Conceptualización	√
Conciencia	√
Construcción de la comunidad	√
Corresponsabilidad	√
Cura	√
Empatía	√
Escucha activa	√
Persuasión	√
Prospectiva	√

Respuesta D. El equipo de desarrollo de Scrum (Proyecto A) debería contar entre 3 y 9 personas. Los equipos de Lean Kanban suelen ser más grandes dependiendo del tamaño de los límites de WIP (Work In Progress) y del número de personas necesarias para acelerar cada entregable con un desperdicio mínimo. / (Principios y mentalidad ágil)

Respuesta A. En un marco ágil, la corrupción del alcance es un problema causado por la inyección de trabajo nuevo o no planificado en medio de una iteración. Las metodologías ágiles resuelven esto a través de procesos. Por ejemplo: el trabajo nuevo solo debe introducirse durante la planificación de la iteración; si el nuevo trabajo tuviera prioridad sobre el actual implica la terminación anticipada de la iteración actual y un regreso a la planificación de la iteración. / El dueño del producto debe priorizar el nuevo trabajo para el proyecto en colaboración con las partes interesadas. O sea, los desarrolladores no realizan la priorización del listado de trabajo pendiente. / En las metodologías ágiles se agrega o modifica alcance permanentemente, pero no durante una iteración.

Respuesta B. El dueño del producto tiene un profundo conocimiento en el área comercial, es responsable de tomar las decisiones relacionadas con el negocio y decide qué historias de usuario (o funcionalidades) se llevarán a cabo en cada iteración. Por lo tanto, el dueño del producto es el responsable de responder a los interesados preguntas tales como: ¿Por qué eliminaron funcionalidades de la iteración ?, ¿Cuáles son las historias que se desarrollarán en la próxima iteración?, ¿Cuál es el detalle de gastos de las funcionalidades?, etc.

Capítulo 2
Entrega orientada al valor

2.1 Entregas incrementales

En proyectos ágiles se busca entregar valor al cliente la más rápido posible. Para ello, en lugar de intentar entregar un producto final con todas las funcionalidades, se suele trabajar con entregas parciales de menor alcance. Por ejemplo:

- Producto viable mínimo (MVP)
- Características comercializables mínimas (MMF)

Producto Viable Mínimo (MVP)

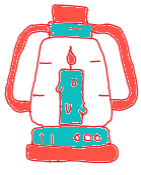
El MVP es la versión preliminar de un nuevo producto que permite al equipo obtener retroalimentación de los clientes de manera anticipada para descubrir si están construyendo el producto correcto.

Mínimo

La base más rudimentaria y básica de la solución posible

Viable

Suficiente para los primeros usuarios



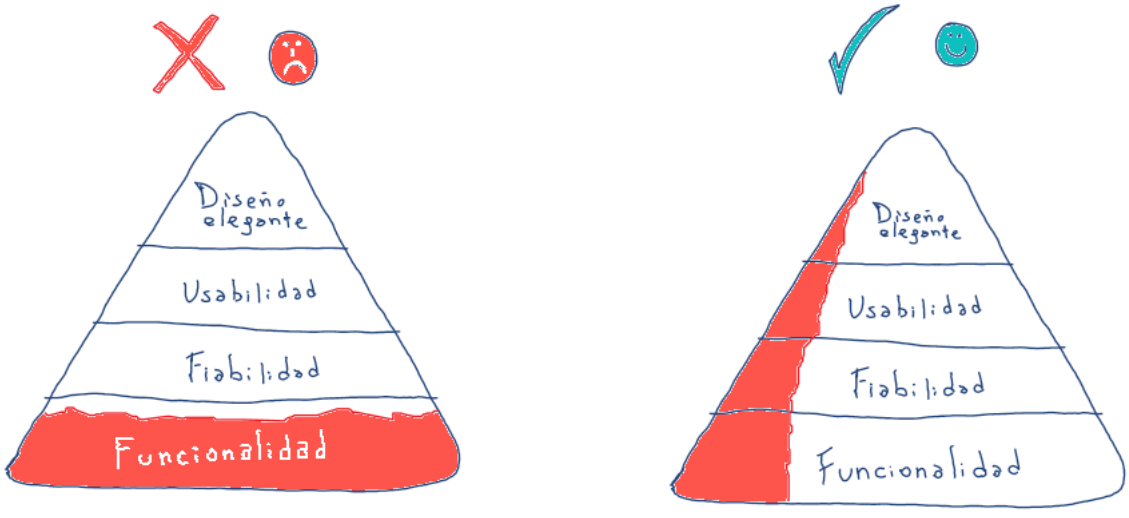
Producto

Algo tangible que los clientes puedan tocar y sentir

El MVP se lleva a cabo utilizando la menor cantidad de tiempo y dinero posible. Aprender de un MVP es más barato que desarrollar un producto con más características, las cuales aumentan costos y riesgo si el producto falla.

Los desarrolladores típicamente lanzarán el MVP para un subconjunto de los posibles clientes, como los "primeros seguidores", que son más tolerantes, más propensos a dar retroalimentación y capaces de captar la visión del producto a partir de un prototipo temprano. O sea, no lanzan el MVP a toda la población de clientes.

El MVP no puede estar basado solamente en las funcionalidades del producto, sino que debe incluir también algo de fiabilidad, usabilidad y diseño.



Con un MVP se busca minimizar el tiempo total gastado en una iteración. El proceso se itera hasta que se obtiene un producto que se ajusta al mercado, o se llega a la conclusión de que el producto no es viable.

El MVP a veces no es un producto en funcionamiento, sino un prototipo preliminar. Por ejemplo, la presentación en ferias internacionales de prototipos de automóviles que aún no funcionan para rastrear los intereses sobre las características preferidas de los potenciales consumidores.

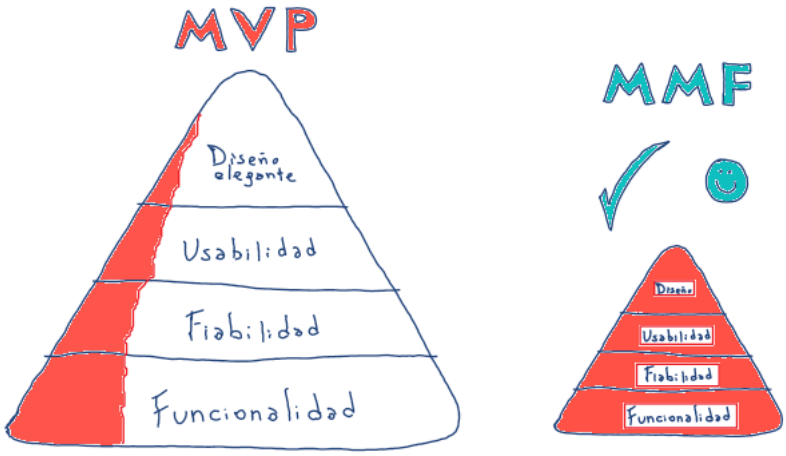
MVP incluye		MVP no incluye
✓ Probar nuevas ideas para aprender	≠	Diseño para obtener beneficios de manera inmediata ✗
✓ Enfoque para resolver un problema	≠	Diseño para impresionar al usuario ✗
✓ Diseño para obtener una rápida retroalimentación con baja inversión	≠	Producto final ✗

Característica comercializable mínima (MMF)

Otra forma de establecer prioridades sobre el alcance inicial del proyecto es definiendo las característica comercializable mínima (MMF: Minimum Marketable Feature).

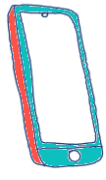
Si bien los conceptos MVP y MMF se suelen utilizar como sinónimos, no significan lo mismo.

MMF es la unidad de funcionalidad más pequeña posible que proporciona un valor tangible a los clientes. A diferencia del MVP que podría ser un prototipo preliminar, MMF siempre es un producto terminado.

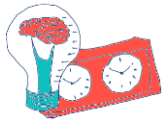


MMF aborda una necesidad específica, resuelve un problema determinado y es de alta calidad y facilidad de uso. Es una característica que se puede comercializar, vender y enviar.

Un ejemplo de MMF es lanzar un celular con un software que incluye las principales funcionalidades y luego agregar gradualmente características adicionales durante la marcha mediante actualizaciones de ese software; en lugar de crear un producto masivo con varias características de una sola vez que luego no son valoradas por el cliente.



Otro ejemplo de MMF sería este libro “Profesional Ágil” que cuando el primer capítulo estuvo terminado, ya estaba a la venta en el mercado, para seguir sumando capítulos a medida que se iban completando.

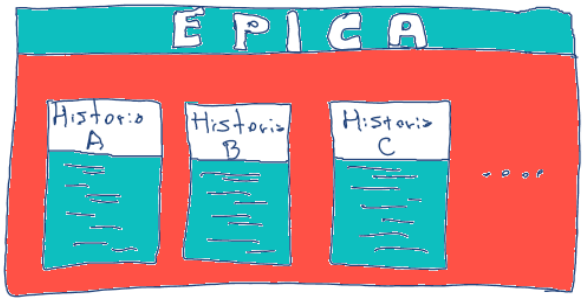


MMF prioriza las características de gran valor y menor tiempo de producción y comercialización, para lanzar productos más rápido al mercado y así aumentar el retorno de la inversión.

2.2 Priorización de los requisitos

En la gestión ágil de proyectos, el alcance del producto inicial es definido por el cliente o su representante (dueño del producto) en forma de requisitos, funcionalidades, épicas o historias de usuario.

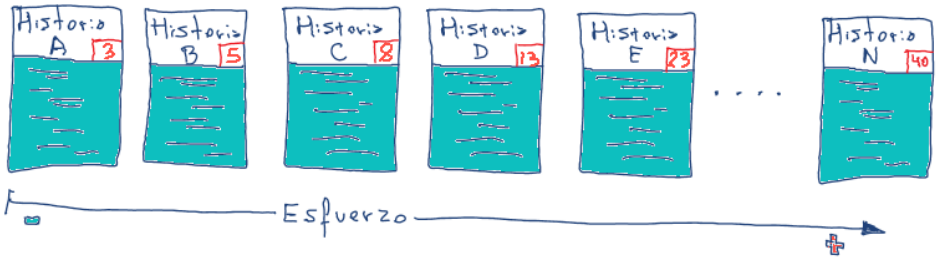
Las épicas son una gran historia de usuario que no puede completarse en una iteración. Por lo tanto, las épicas se dividen en historias de usuario más pequeñas las cuales podrán ser entregadas en distintas iteraciones.



Las historias de usuario son una breve descripción escrita que representa un requisito o funcionalidad del proyecto utilizando el lenguaje común del usuario.

Todas las historias de usuario forman parte del trabajo pendiente asociado al producto (product backlog). Las historias de usuario seleccionadas para desarrollar durante las iteraciones definen el alcance del proyecto.

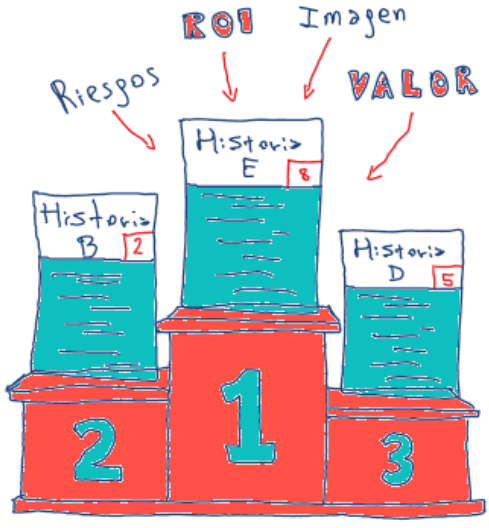
El esfuerzo para realizar cada historia de usuario podría ser estimado en horas. Sin embargo, el equipo de desarrollo suele agregar puntos de historia (story-points) a cada historia de usuario según el esfuerzo relativo necesario para completarlas. Estos puntos de historia no tienen relación con horas de trabajo. Por ejemplo, a una historia de mínimo esfuerzo se le podrían asignar 3 puntos, otra historia de esfuerzo medio 23 puntos y una historia de bastante esfuerzo podría tener 40 puntos de historia.



Con el objetivo de entregar valor al cliente lo antes posible, las historias de usuarios deben ser priorizadas por el cliente (o dueño del producto) de acuerdo con su valor, no solo por el esfuerzo que requieran.

El dueño del producto y las personas vinculadas al negocio son quienes conocen mejor las historias que aportarán más valor. Para estimar el valor de cada historia se tendrán en cuenta factores tales como:

- Generar un mayor retorno de la inversión
- Resolver más problemas
- Optimizar los procesos más importantes
- Resolver problemas tecnológicos para poder avanzar con el desarrollo del producto
- Reducir riesgos
- Mejoras de la imagen empresarial
- Incrementar la participación de mercado
- Esfuerzo necesario o costos asociados a cada historia

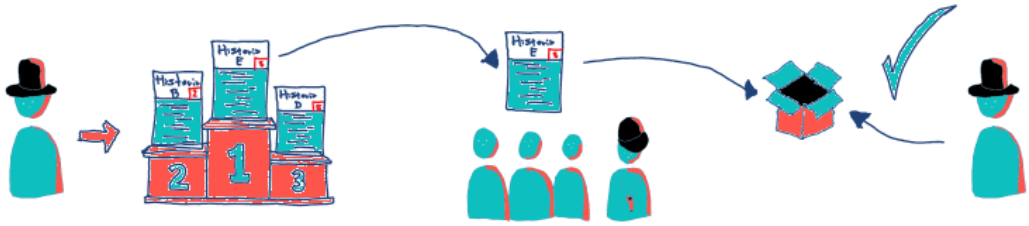


Si bien el dueño del producto es el principal responsable de priorizar las historias de usuario, el equipo técnico suele colaborar activamente en esta priorización, estimando el esfuerzo necesario de cada historia de usuario, identificando los riesgos tecnológicos, definiendo las dependencias, etc.

Una vez priorizadas las historias según su valor, el equipo de desarrollo comenzará a desarrollarlas teniendo en cuenta esas prioridades.

El equipo de desarrollo será quien defina cómo llevar a cabo cada historia de usuario, planificando el trabajo, los enfoques y técnicas a utilizar.

Por último, cuando el equipo entrega el incremento del producto, el cliente verifica si ese entregable agregó el valor estimado en función de las necesidades y expectativas del usuario.



A continuación, veremos algunas técnicas cuantitativas para priorizar los requisitos, funcionalidades, épicas o historias de usuario, en función del valor monetario que se estima podrían generar.

ROI

VAN

TIR

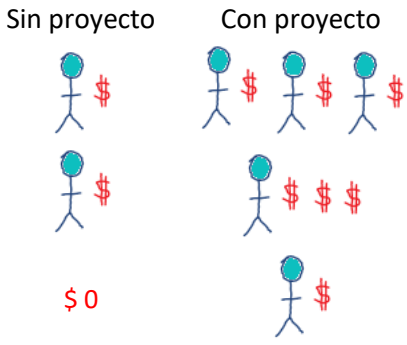
Retorno de la inversión (ROI)

Las entregas incrementales del proyecto por lo general van a generar algún tipo de ingresos o ahorro de costos a la organización. Los ingresos podrían clasificarse en:

Nuevos: obtenidos por nuevos clientes

Incrementales: obtenidos de los clientes existentes al agregar nuevas funcionalidades al producto

Retenidos: se perderán los clientes actuales si no se desarrolla una nueva característica



En la tabla a continuación se presenta un ejemplo con los ingresos generados por cinco épicas en las cuáles es necesario invertir $100.

Flujo Fondos	Hoy	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Total	ROI
Épica A	-$ 100	$ 0	$ 0	$ 25	$ 40	$ 50	$ 15	15%
Épica B	-$ 100	$ 55	$ 40	$ 5	$ 5	$ 5	$ 10	10%
Épica C	-$ 100	$ 0	$ 0	$ 0	$ 0	$ 117	$ 17	17%
Épica D	-$ 100	$ 20	$ 20	$ 20	$ 20	$ 25	$ 5	5%

El retorno de la inversión (ROI: return on investment) del flujo de fondos de la Épica A sería de 15% en 5 meses ($15 de ganancia neta / $100 de inversión inicial).

$$ROI = \frac{\text{Ingresos netos}}{\text{Inversión}}$$

Si observamos el ingreso neto o ROI que genera cada historia, la mejor sería C ($17, 17%) y la peor D ($5, 5%). Sin embargo, esos valores no tienen en consideración el valor del dinero en el tiempo, por lo que el ROI tiene sus limitaciones para seleccionar entre la mejor alternativa de inversión.

A continuación, mostraremos los criterios de selección de VAN y TIR que tienen en cuenta el valor del dinero en el tiempo.

Valor actual neto (VAN)

Mientras antes se obtengan los ingresos, más beneficioso será para el proyecto.

Para calcular el valor actual de un ingreso futuro hay que considerar el valor del dinero en el tiempo. O sea, $100 mañana valen menos que $100 hoy. El costo de oportunidad del dinero podría ser el retorno de invertir en otros proyectos de riesgo similar.

Supongamos, por ejemplo, que el costo de oportunidad del dinero fuera de 1% mensual y que un lanzamiento del proyecto generará un valor de $100.000 dentro de 3 meses. Ese dinero equivaldría a como si hoy hubiéramos tenido un ingreso de $97.059.

Valor Actual = Valor Futuro / $(1+i)^n$ = $100.000 / $1{,}01^3$ = $97.059

Supongamos ahora que para desarrollar una épica tenemos que invertir $100 hoy para obtener ingresos de $30 en el mes 3, $40 en el mes 4 y $50 en el mes 5. Para saber si es rentable o no esa inversión, tenemos que calcular el valor actual neto.

	Hoy	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Total
Flujo Fondos	-$ 100	$ 0	$ 0	$ 30	$ 40	$ 50	**$ 20**
Valor actual	-$ 100	$ 0	$ 0	$ 29,12	$ 38,44	$ 47,57	**$ 15,13**

El valor actual neto de esa inversión es la sumatoria de todos los valores actuales, en este ejemplo $15,13 (-100 + 29,12 + 38,44 + 47,57) utilizando una tasa de descuento del 1% mensual. Si el VAN es positivo, es rentable realizar esa inversión.

Para priorizar entre distintas épicas que generarán ingresos en diferentes momentos del tiempo, podríamos utilizar el concepto del VAN.

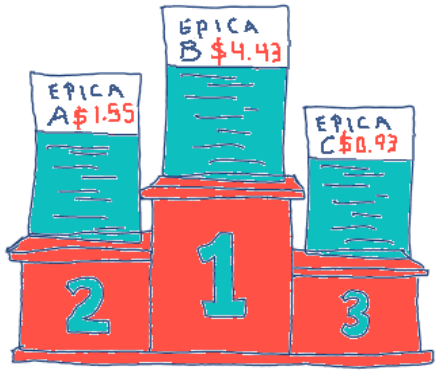
Supongamos que tenemos que seleccionar entre las mismas cuatro épicas presentadas previamente:

Flujo Fondos	Hoy	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Total	ROI
Épica A	-$ 100	$ 0	$ 0	$ 25	$ 40	$ 50	$ 15	15%
Épica B	-$ 100	$ 55	$ 40	$ 5	$ 5	$ 5	$ 10	10%
Épica C	-$ 100	$ 0	$ 0	$ 0	$ 0	$ 117	$ 17	17%
Épica D	-$ 100	$ 20	$ 20	$ 20	$ 20	$ 25	$ 5	5%

Si el costo de oportunidad del dinero fuera, por ejemplo, un 3% mensual, en la tabla a continuación se presenta el valor actual neto de cada alternativa.

	Valor actual						VAN
	Hoy	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Total
Épica A	-$ 100	$ 0	$ 0	$ 22,88	$ 35,54	$ 43,13	$ 1,55
Épica B	-$ 100	$ 53,40	$ 37,70	$ 4,58	$ 4,44	$ 4,31	$ 4,43
Épica C	-$ 100	$ 0	$ 0	$ 0	$ 0	$ 100,93	$ 0,93
Épica D	-$ 100	$ 19,4	$ 18,9	$ 18,3	$ 17,8	$ 21,6	- $ 4,09

A mayor VAN, mayor prioridad tendrá la épica para su inicio de desarrollo. O sea, en base al VAN el criterio para fijar prioridades sería 1º B, 2º A y 3º C.



La épica D, al tener un VAN negativo (-$4,09), no debería ser una opción para desarrollar desde el punto de vista económico.

Tasa Interna de Retorno (TIR)

Otra forma de priorizar las épicas sería comparando la **tasa interna de retorno (TIR)** de cada alternativa. La TIR es la tasa de interés que hace el valor actual neto igual a cero.

Siguiendo con el mismo ejemplo, en la tabla se presenta la TIR de cada alternativa:

Flujo Fondos	Hoy	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	TIR	Prioridad
Épica A	-$ 100	$ 0	$ 0	$ 25	$ 40	$ 50	**3,38%**	**2º**
Épica B	-$ 100	$ 55	$ 40	$ 5	$ 5	$ 5	**5,62%**	**1º**
Épica C	-$ 100	$ 0	$ 0	$ 0	$ 0	$ 117	**3,19%**	**3º**
Épica D	-$ 100	$ 20	$ 20	$ 20	$ 20	$ 25	**1,60%**	

No te preocupes por intentar calcular las TIR manualmente, ya que eso no lo harás jamás en la vida real y tampoco lo necesita para rendir la certificación PMI-ACP®. ¡Podrías utilizar la fórmula TIR en Excel y listo! ☺

A los fines de demostrar que la TIR es la tasa de descuento que hace el VAN igual a cero, en la tabla a continuación descontamos el flujo de fondos de cada alternativa a las TIR.

Valor actual	Hoy	1	2	3	4	5	Total = VAN
Épica A	-$ 100	$ 0	$ 0	$ 22,63	$ 35,02	$ 42,35	**$ 0**
Épica B	-$ 100	$ 52,07	$ 35,86	$ 4,24	$ 4,02	$ 3,80	**$ 0**
Épica C	-$ 100	$ 0	$ 0	$ 0	$ 0	$ 100	**$ 0**
Épica D	-$ 100	$ 19,69	$ 19,38	$ 19,07	$ 18,77	$ 23,10	**$ 0**

Cuando la TIR es mayor que el costo de oportunidad del dinero, el VAN es positivo. En nuestro ejemplo, la TIR de A (3,38%), B (5,62%) y C (3,19%) son mayores a la tasa de descuento del 3%. Por lo tanto, el VAN de A, B y C son positivos.

Por otro lado, si la TIR es inferior al costo del dinero, el VAN es negativo, como ocurre con la historia D (TIR 1,6% menor a tasa de descuento del 3%).

Resumiendo, la TIR de cada alternativa también se utiliza como un criterio de priorización. A mayor TIR mayor prioridad debería tener la épica a desarrollar. Si quisiera profundizar los problemas de la TIR para seleccionar entre alternativas, le recomiendo el libro "Evaluación Financiera de Proyectos" de Pablo Lledó.

2.3 Técnicas de Priorización

Si bien el ROI, VAN, o TIR deberían ser los criterios más utilizados de priorización, en la práctica suele ocurrir que no se cuenta con la información para poder estimar el flujo de fondos que podría generar cada épica o historia de usuario. Además, invertir demasiado tiempo en conseguir esa información podría ser poco eficiente en proyectos ágiles. Por lo tanto, a continuación, presentaremos otros métodos cualitativos de priorización.

Método de los 100-puntos

El método de los 100 Puntos (o 100 dólares o Monopoly Money) se utiliza para priorizar las historias de usuario del listado de trabajo pendiente en función del valor que agregan al negocio.

A cada interesado se les da 100 puntos (o $100) para que puedan distribuir entre las historias de usuario en función de lo que consideran que agregará más valor.

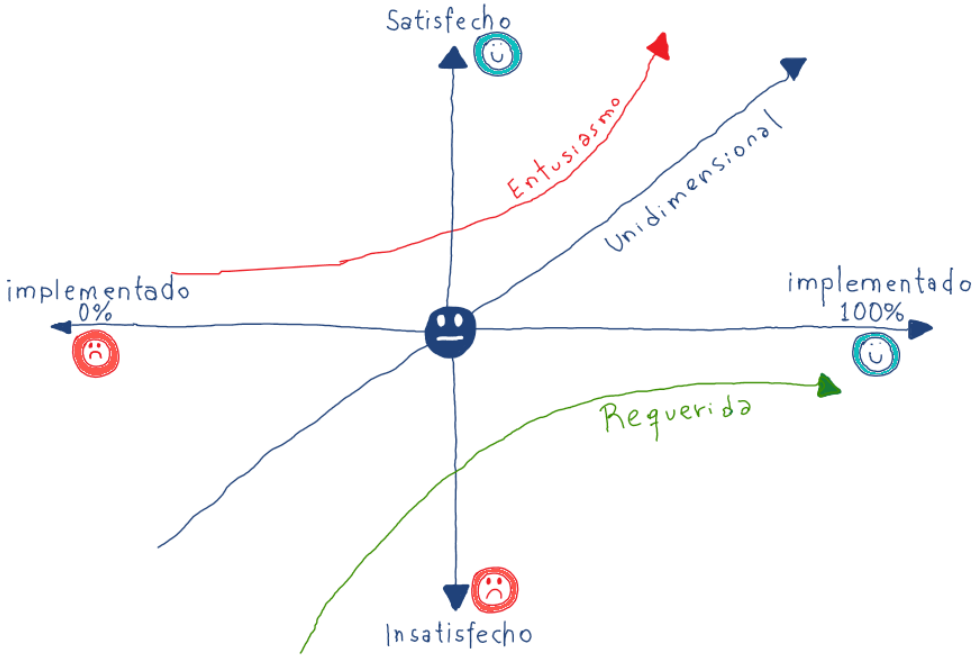
Por ejemplo, en la tabla a continuación, Roberto decidió que las cinco historias tienen igual importancia, asignando $20 a cada una. María decidió que la historia A es lo más importante y que E no agrega valor. Por último, Nurg considera que B es lo más importante.

	A	B	C	D	E	TOTAL
Roberto	$20	$20	$20	$20	$20	$100
María	$40	$30	$20	$10	$0	$100
Nurg	$20	$40	$30	$10	$0	$100
TOTAL	$80 2°	$90 1°	$70 3°	$40	$20	

Cada persona distribuye sus $100 entre las historias, luego se suman los valores de todos los participantes para obtener la lista de historias priorizadas. Siguiendo con el ejemplo, la historia de usuario más importante sería B, luego A y después C.

Análisis Kano

El modelo Kano es una teoría de desarrollo de productos y de satisfacción del cliente que clasifica a las preferencias del cliente en cinco categorías o atributos.



1. Calidad requerida (Must-be Quality): no generan satisfacción cuando se cumplen porque son atributos básicos, pero dan lugar a gran insatisfacción cuando no se cumplen.

2. Calidad unidimensional o de resultado: dan como resultado la satisfacción cuando se cumplen e insatisfacción cuando no se cumplen porque fueron ofrecidos por el vendedor.

3. Calidad de entusiasmo o atractiva (Delighters): proporcionan satisfacción cuando se logran plenamente, pero no causan insatisfacción cuando no se logran porque no son esperados por el cliente ni fueron mencionados antes de la compra.

4. Calidad indiferente: aspectos que no son ni buenos ni malos; no generan satisfacción o insatisfacción del cliente.

5. Calidad inversa: un alto grado de atributos resulta en la insatisfacción de algunos clientes. Por ejemplo, algunos clientes prefieren los productos de alta tecnología con muchas funcionalidades, mientras que otros quedan insatisfechos si un producto tiene demasiadas funcionalidades tecnológicas que les complican el uso.

Con el modelo Kano clasificamos los requisitos según los tipos de calidad para luego priorizarlos. Por ejemplo, la calidad requerida y unidimensional suele ser lo más prioritario.

MoSCoW

Categorizar los diferentes requisitos del proyecto desde los que son estrictamente necesarios (prioritarios) hasta los que no serían necesarios por el momento (no prioritarios). Las letras MoSCoW se derivan de los siguientes términos del inglés:

Must have (obligatorio)

Should have (deberían estar)

Could have (podría implementarse)

Won't have (no lo queremos por el momento)

Puño a cinco (fist to five)

El método de votación puño a cinco es una técnica utilizada para encuestar a los interesados sobre el valor de las historias de usuario a los fines de lograr un consenso en la priorización.

Cada miembro del equipo responde levantando su mano con un puño o con el número de dedos según el nivel de apoyo.

Puño: desea bloquear la propuesta porque cree que es perjudicial.

Un dedo: tiene reservas serias, pero no bloqueará la iniciativa. Se compromete a abrir la comunicación con respecto a sus reservas.

Dos dedos: tiene reservas importantes, pero apoyará la iniciativa.

Tres dedos: apoya la iniciativa.

Cuatro dedos: brinda un fuerte apoyo y participación, pero no está dispuesto a liderar la iniciativa.

Cinco dedos: lo considera como la mejor idea, tomará la iniciativa si se lo piden.

Modelo de Karl Wiegers

El modelo de priorización de Karl Wiegers, utiliza una escala cualitativa del 1 al 9 para calificar cada uno de los siguientes factores: beneficio, penalidad, costo y riesgo.

- Beneficio. 1 sería que no genera ingresos o no está alineado con los requisitos comerciales del producto, hasta 9 con gran satisfacción del cliente o ROI.
- Penalidad. 1 si no hay penalización y 9 indica un problema muy serio. Por ejemplo, el incumplimiento de una regulación gubernamental podría incurrir en una multa alta, incluso si el beneficio para el cliente es bajo.

= Beneficio + Penalidad. Por ejemplo, los requisitos que tienen un beneficio y penalidad bajos agregan poco valor, por lo que pueden ser casos de "enchapado en oro".

- Costo: gastos o esfuerzo de realizar el desarrollo. 1 implica bajo costo y 9 alto costo.
- Riesgo: amenazas o problemas tecnológicos de implementar la funcionalidad. 1 es de muy bajo riesgo y 9 muy alto riesgo.

= Costo + Riesgo

Los ítems prioritarios son aquellos de alto valor y bajos costos. Una forma de calcular la prioridad es comparando el valor con los costos.

Prioridad = Valor / Costos

Veamos un ejemplo en la tabla a continuación donde los expertos comerciales y el equipo técnico han calificado con puntajes cualitativos del 1 al 9 cuatro historias de usuario.

Historia	Beneficio (a)	Penalidad (b)	**Valor (c)** (a) + (b)	Costo (d)	Riesgo (e)	**Costos (f)** (d) + (e)	**Prioridad** (c) / (f)
A	7	4	11	9	7	16	**0,69**
B	2	1	3	1	1	2	**1,50**
C	5	3	8	2	1	3	**2,67**
D	9	8	17	7	8	15	**1,13**

En este ejemplo las prioridades de desarrollo serán: 1º C, 2º B, 3º D y 4º A.

También se suelen trabajar las prioridades colocando ponderaciones. Por ejemplo, otorgar más ponderación a los beneficios que a la penalidad o más ponderación al costo que al riesgo.

Puntos de valor de negocio

Así como se suelen asignar puntos de historia al esfuerzo relativo que requiere realizar cada épica o historia de usuario, se pueden asignar "puntos de valor de negocio (business value points)" relativos para cada épica.

Por ejemplo, a una épica de poco valor se le podrían asignar 3 puntos, otra épica de valor medio 8 puntos y una épica de mucho valor podría tener 21 puntos.

La priorización entre épicas se puede obtener comparando la relación entre los puntos de valor relativo con los puntos de esfuerzo relativo de cada épica.

Priorización = Puntos de valor / Puntos de esfuerzo

Al momento de priorizar, podríamos ponderar como más importante el valor que el esfuerzo.

En la tabla a continuación se presenta un ejemplo donde se tiene que priorizar entre siete épicas. El dueño del producto agregó puntos de valor relativo a cada épica y el equipo de desarrollo estimó el esfuerzo relativo de cada una de esas épicas. Además, se decidió ponderar en 70% los puntos de valor y en un 30% los puntos de esfuerzo.

Épica	Valor	70% Valor	Esfuerzo	30% Esfuerzo	V / E	Prioridad
A	2	1,4	5	1,5	0,93	**6º**
B	3	2,1	5	1,5	1,40	**5º**
C	5	3,5	2	0,6	5,83	**2º**
D	8	5,6	13	3,9	1,44	**4º**
E	13	9,1	5	1,5	6,07	**1º**
F	21	14,7	21	6,3	2,33	**3º**

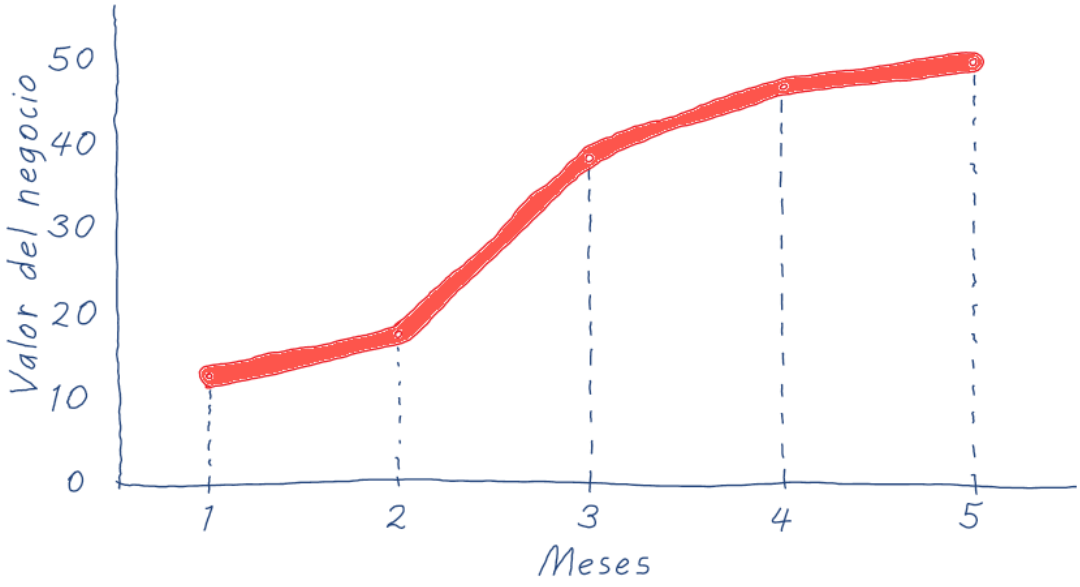
Mientras mayor sea el ratio Valor / Esfuerzo, mayor será la prioridad. O sea, realizar primero las actividades de mayor valor para la menor cantidad de esfuerzo posible.

Una vez priorizadas las historias de usuario, cuando el proyecto esté en marcha, se puede hacer un seguimiento de los puntos de valor del negocio a través del tiempo.

Supongamos que cinco épicas que priorizamos se completaron realizando lanzamientos mensuales como se presenta a continuación.

Épica	Mes de Lanzamiento	Valor	Valor acumulado
E	1	13	13
C	2	5	18
F	3	21	39
D	4	8	47
B	5	3	50

Si graficamos el valor acumulado a través del tiempo obtenemos el siguiente diagrama de flujo acumulativo que nos indica como evoluciona el valor del proyecto.

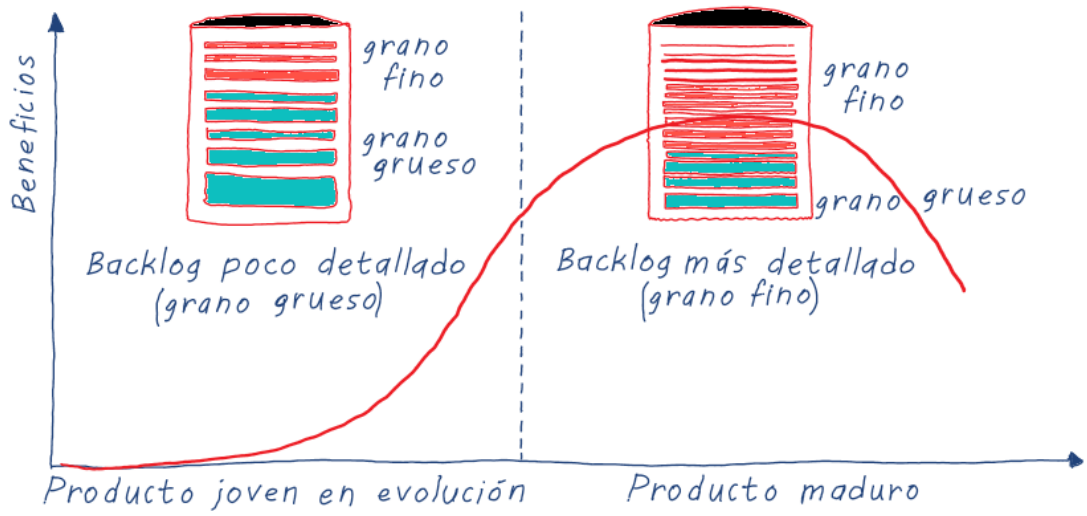


2.4 Especificación de los requisitos

Un product backlog excesivamente detallado o de grano fino (fine grained) será difícil de gestionar. Pero si es demasiado agregado o de grano grueso (coarse grained), tampoco será útil porque proporciona muy poca dirección al equipo de desarrollo.

Las historias de usuario del product backlog se ordenan con los elementos de alta prioridad en la parte superior y los elementos de baja prioridad en la parte inferior. Los elementos prioritarios, aquellas historias que se van a realizar en la próxima iteración, deben ser de grano fino, mientras que a medida que disminuye su prioridad, deberían volverse cada vez más de grano grueso a nivel de épicas con historias de usuario con información incompleta.

Por otro lado, el nivel de detalle del product backlog dependerá del ciclo de vida del proyecto. En las fases iniciales cuando el producto es joven, el product backlog suele tener menor nivel de detalle relativo comparado con fases avanzadas donde el proyecto ya está más maduro. [11]



Especificación mediante ejemplos (SBE: Specification by Example): enfoque colaborativo para definir los requisitos y las pruebas funcionales basado en ejemplos realistas, en lugar de enunciados abstractos.

[11] Roman Pichler, www.romanpichler.com

2.5 Entregas rápidas y flexibles

Todo proyecto ágil persigue el triple objetivo de ser “rápido, flexible y fluido”.

Rápido (Fast): para que los entregable lleguen al cliente lo más rápido posible es necesario eliminar los desperdicios, las burocracias y la sobreproducción trabajando con lotes pequeños en ciclos incrementales de desarrollo manteniendo relaciones estrechas entre los interesados.

Flexible: el desarrollo del producto debe permitir la customización según las necesidades del cliente para poder gestionar los cambios que sean necesarios.

Fluido (Flow): entregar incrementos del producto de manera continua.

Una de las formas de realizar entregas rápidas y fluidas es aplicando el principio Lean que dice “que el valor fluya sin interrupciones”. Para ello será fundamental comprender el concepto de tiempo valor agregado y eliminar cualquier desperdicio que afecte el flujo de producción.

Tiempo valor agregado

El tiempo valor agregado es aquel que mejora el resultado de un proceso que suele incluir solamente el tiempo de desarrollo o producción.

Por ejemplo, supongamos que para la producción de un entregable necesitamos 5 horas de trabajo. Sin embargo, hay otras actividades relacionadas con esa producción como: traslados de inventarios (10hs), espera para disponibilidad de recursos (20hs), movimientos entre las estaciones de trabajo (15hs) y el almacenamiento antes de entregar al cliente (30hs).

En este ejemplo, se considera tiempo valor agregado solamente las 5 horas de desarrollo, todo el resto (75 horas) no es tiempo valor agregado, por lo que son desperdicios que debería intentar eliminar o disminuir.

Ratio valor agregado = tiempo valor agregado / tiempo total desde la recepción hasta entrega.

Tiempo total = 5 + 10 + 20 + 15 + 30 = 80 horas

Ratio valor agregado = 5 / 80 = 6,2%

Eliminación de desperdicios

Desperdicio es todo aquello que no agrega valor al proyecto desde el punto de vista del cliente. Los siete desperdicios de la producción se pueden resumir con el acrónimo en inglés TIMWOOD.

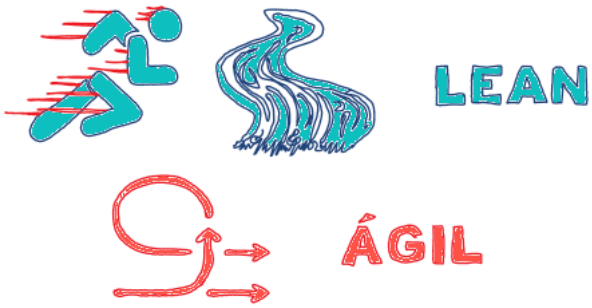
1. Transporte: el traslado innecesario de productos, equipos y personas resulta en tiempo perdido y aumenta la probabilidad que los materiales se dañen.
2. Inventario: el inventario excesivo puede llevar a tiempos de entrega más largos, productos dañados, uso ineficiente del capital, menor capacidad de almacenamiento disponible y costos adicionales para administrarlo.
3. Movimiento: los movimientos innecesarios, peligrosos y complicados pueden causar daños a los empleados, al equipo y al producto.
4. Waiting and Delay (Esperas y retrasos): tiempo que se pierde debido a una espera en la productividad (ej. mantenimiento inesperado del equipo).
5. Overproduction (Sobreproducción): cuando la producción excede la demanda del cliente, las instalaciones se quedan con un inventario excesivo para administrar.
6. Overprocessing (Sobre procesamiento): cuando el producto o servicio tiene más características que las requeridas, solo agrega costos y riesgos.
7. Defectos: productos que no cumplen con los estándares de la compañía que deben desecharse o reelaborarse, lo que agrega costos a la operación sin agregar valor al cliente.

Mary y Tom Poppendieck en su libro Lean Software Development describen los siete desperdicios del desarrollo de software.

Desperdicios del desarrollo de software		Cómo mitigarlo
T	Trabajo parcialmente realizado o en proceso	Colocar límites del WIP
I	Entregar características adicionales innecesarias	Validar supuestos sobre el valor en ciclos cortos de retroalimentación
M	Reaprendizaje (aprender lo mismo)	Aprendizaje compartido
W	Traspasos o transferencias	Equipos autónomos y multifuncionales
O	Retrasos	Reducir las dependencias entre los equipos al unir el trabajo relacionado
O	Cambiar de contexto	Colocar límites del WIP y definir prioridades
D	Defectos	Desarrollo basado en pruebas

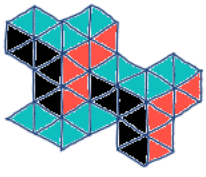
Fabricación ágil (Agile Manufacturing)

Si bien los principios Lean ayudan a planificar entregas rápidas y fluidas, es la filosofía ágil la que agrega el concepto de "flexibilidad" para adaptar el proyecto a la necesidades cambiantes del contexto para satisfacer rápidamente las demandas de los clientes.



En la fabricación ágil, se utiliza tecnología y sistemas de comunicación altamente integrados para desarrollar productos de excelente calidad que se hacen de forma rápida y rentable.

Los cuatro pilares de la fabricación ágil son:



Diseño de producto modular: permite una variación rápida y fácil.

Tecnología de la información: permite la comunicación entre departamentos y una respuesta rápida a los pedidos.

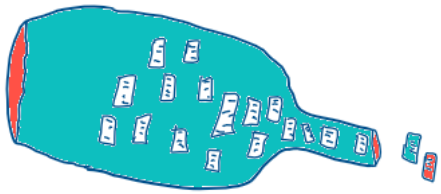
Socios corporativos: formar alianzas con socios estratégicos para mejorar el tiempo de comercialización de ciertos productos

Cultura del conocimiento: la capacitación de los empleados ayuda a respaldar un cambio rápido y continuo en toda la empresa.

Cada uno de estos cuatro pilares sirve para mejorar el producto final, así como para garantizar que el cliente esté completamente satisfecho.

2.6 Trabajo en progreso (WIP)

En todo proyecto existen restricciones o cuellos de botella al flujo de valor. Cualquier intento de optimizar el flujo del proyecto sin mejorar la restricción, será un desperdicio, ya que no mejorará el rendimiento general.

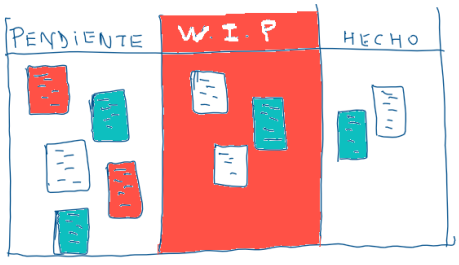


En los proyectos ágiles, los esfuerzos de la organización deben dirigirse a mejorar la restricción hasta que deje de ser una restricción. Por ejemplo, colocar límites máximos del trabajo en progreso (WIP) en los tableros Kanban.

Las restricciones del proyecto también podrían originarse en los miembros del equipo. Por ejemplo, si una persona desempeña un papel crítico en un proceso, el flujo de valor del proyecto no podrá avanzar más rápido de lo que esa persona pueda procesar. Además, existe el riesgo de una interrupción completa del proyecto si esa persona deja de trabajar por algún motivo. Estas personas no son héroes, sino una amenaza para el proyecto.

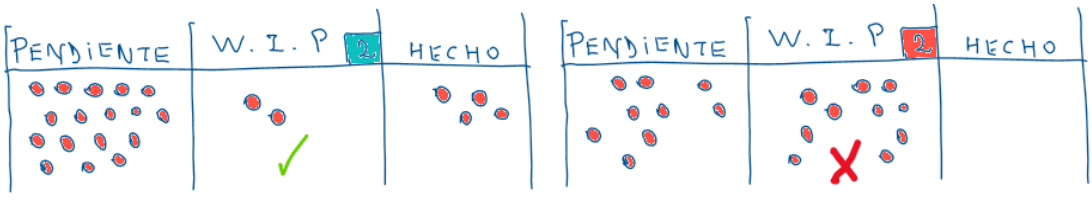
Por lo tanto, se busca siempre que los equipos ágiles sean multifuncionales para que el proyecto no dependa de una sola persona. De esta forma se reducen los cuellos de botella y mejora el flujo de valor del proyecto.

El **trabajo en progreso (WIP: work in progress)** representa todas las actividades en las que el equipo está trabajando que deberán completarse para entregar valor al cliente. No se consideran en el WIP aquellas tareas que no han comenzado o que ya están finalizadas.

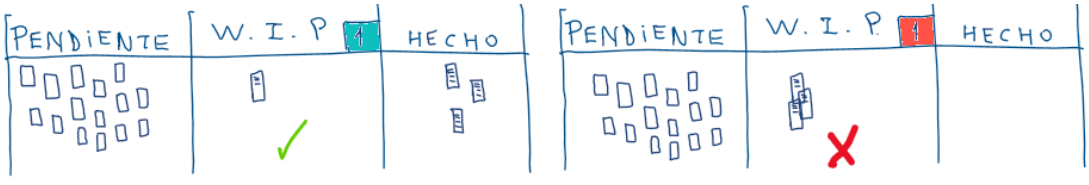


El principio básico detrás de Lean-Kanban es el de reducir el WIP para aumentar el ritmo de producción y flujo de valor que llega al cliente. La idea es que el equipo finalice las tareas para el cliente antes de comenzar con nuevas actividades.

Por ejemplo, supongamos que una persona nos lanza pelotitas de ping pong y nuestro trabajo es atraparlas y colocarlas dentro de una caja. Si nos lanza una pelotita a la vez podríamos guardar una pelotita por segundo; si somos muy dúctiles tal vez podríamos atrapar hasta un máximo de 2 pelotitas (una con cada mano) y guardarlas en un segundo; pero si nos lanzan 10 pelotitas juntas, lo más probable es que se nos caigan casi todas, demoremos mucho tiempo en recogerlas, tengamos que volver a comenzar el trabajo y aumentar bastante el plazo de producción, donde el guardado de cada pelotita podría extenderse a minutos. O sea, en este ejemplo, el máximo WIP sería de 2.



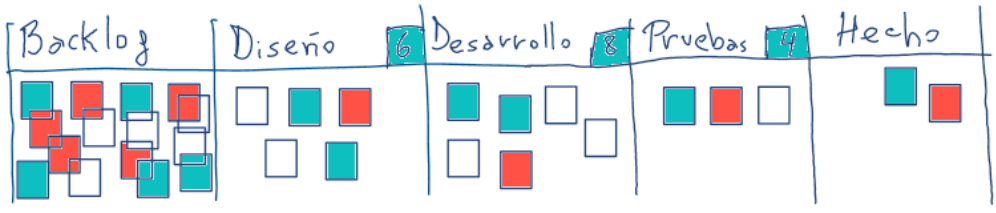
Otro ejemplo podría ser el de una impresora, donde el máximo WIP de páginas en impresión sería 1 para imprimirlas lo más rápido posible . Si queremos imprimir 2 o más páginas al mismo tiempo, el papel se atascaría y demoraríamos más tiempo corrigiendo los problemas.



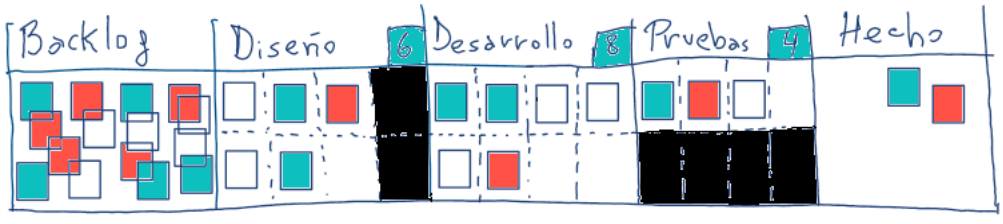
Límites del WIP

No hay una fórmula predeterminada que nos diga cuál debería ser el límite del WIP, esto se basa generalmente en la experiencia, tipo de proyecto, cultura organizacional, pruebas y error, etc. Además, el flujo de trabajo cambiará dinámicamente por lo que deberá adaptar los límites WIP en función de los factores cambiantes, como los nuevos requisitos comerciales, la demanda del mercado, el tamaño y las habilidades del equipo, problemas técnicos, etc.

Alguna organizaciones comienzan con un WIP que surge de multiplicar por 2 la cantidad de personas que están trabajando en una actividad. Por ejemplo, supongamos un proyecto que tiene tres actividades secuenciales: Diseño-Desarrollo-Pruebas. Si tenemos 3 diseñadores, 4 desarrolladores y 2 probadores, el WIP de cada una de esas actividades será 6, 8 y 4 respectivamente.



Una forma de limitar el WIP en tableros Kanban físicos (low tech – high touch) es limitando el espacio físico donde se puedan agregar las tarjetas con el trabajo a realizar. De esa forma, si alguien quisiera agregar una tarjeta en algún casillero que ya está ocupado, rápidamente nos daríamos cuenta de que se ha excedido el límite del WIP.

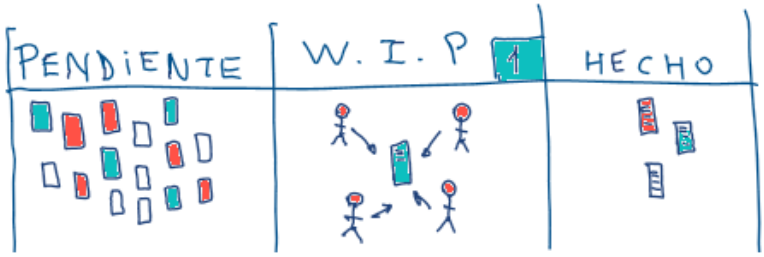


En software con tableros Kanban digitales, se puede fijar el WIP en cada actividad y cuando algún miembro del equipo quiere agregar trabajo en exceso al límite del WIP, aparece un error en pantalla.

Otras forma para definir el WIP la primera vez es con la fórmula [(2 x n) – 1], donde n es el número de miembros del equipo y el -1 se aplica para aumentar el grado de colaboración. O sea, en un equipo de 7 personas, el límite del WIP sería 13. Pero esto no es más que un consejo práctico, ya que cada equipo de proyecto debe identificar su propio WIP y adaptarlo a la dinámica del contexto.

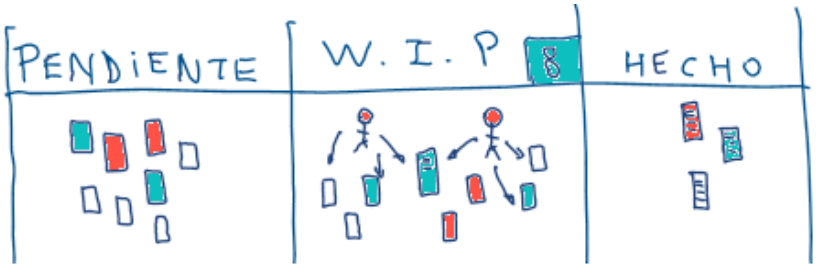
Tamaño del WIP

Si el límite del WIP fuera demasiado bajo, por ejemplo, un WIP de 1 para un equipo de 4 personas, esa tarea se podría realizar muy rápido ya que tendrá 4 personas trabajando solamente en esa actividad hasta terminarla.



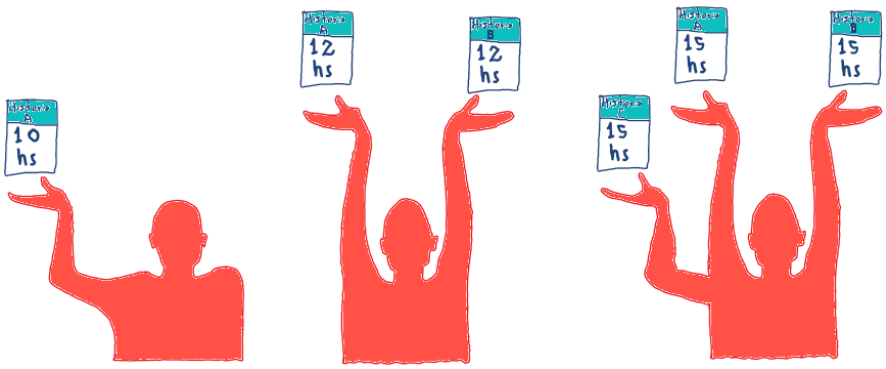
Si bien un WIP bajo fomenta la colaboración y trabajo en equipo, es probable que tengamos miembros del equipo ociosos porque no podrán trabajar juntos todo el tiempo sobre esa misma tarea. Además, cuando el WIP es muy bajo, tendremos varias tareas en estado de "pendientes" demasiado tiempo, en espera de que el equipo comience a trabajar sobre ellas. Por lo tanto, el "tiempo de entrega" desde que el cliente realiza el pedido hasta que se entrega el producto, será mayor de lo necesario.

En el otro extremo, si el límite del WIP es demasiado alto, por ejemplo, un WIP de 8 para un equipo de 2 personas, existe el riesgo de empezar muchas tareas y terminar pocas. Frente a cualquier contratiempo en una tarea, el equipo, en lugar de enfocarse en resolver el problema y terminar la tarea, podrán optar por comenzar con otra tarea y demorar las que presenten problemas.

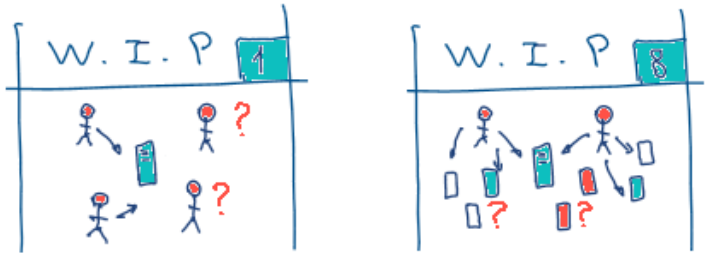


El trabajo en multitareas con el ciclo “arrancar-frenar-arrancar-frenar…” no ayudarán en la productividad y eficiencia del trabajo. El cambio de tareas implica un cambio en el estado mental para recrear el ambiente, el conocimiento y el ritmo necesario para trabajar en cada actividad.

Por ejemplo, Jeff Atwood menciona que se puede perder hasta el 20% del tiempo de un programador debido al cambio de contexto cuando se agrega un segundo proyecto a su lista y casi un 50% cuando se agrega un tercer proyecto.



Resumiendo, si observamos miembros del equipo ociosos sin estar trabajando en tareas es probable que el WIP sea demasiado bajo. Y si observamos tareas en el listado del WIP pero que no tienen personas trabajando en ellas, es probable que el WIP sea demasiado alto.

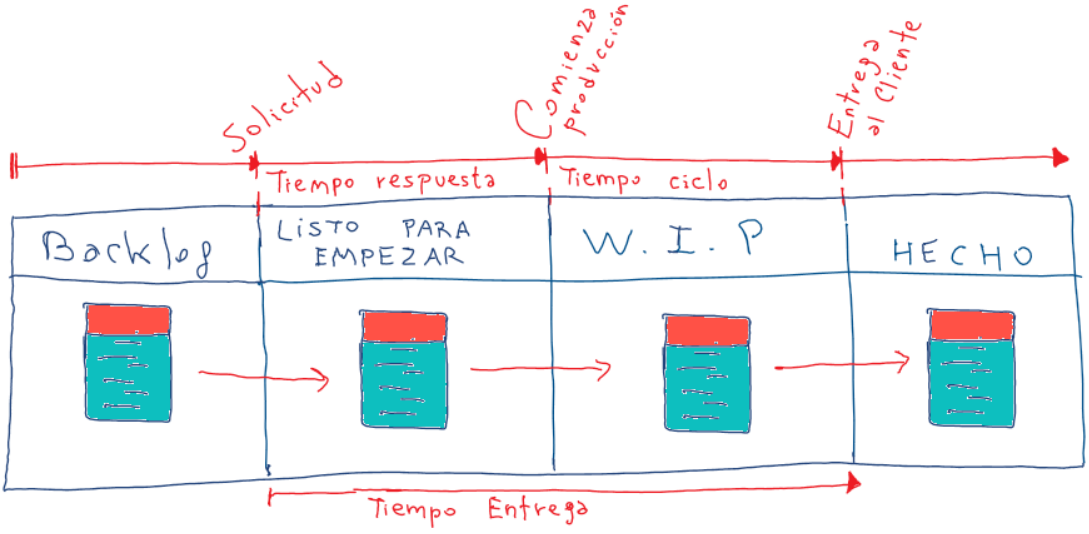


Al aplicar los límites al WIP, el equipo tendrá la oportunidad de localizar los cuellos de botella en los procesos de trabajo antes de que éstos se conviertan en bloqueos que retrasan la entrega de valor al cliente.

2.7 Tiempo de entrega y ciclo

Para entregar valor al cliente lo antes posible es necesario gestionar los tiempos y la productividad del proceso de desarrollo. Los conceptos más importantes son:

- **Tiempo de respuesta (response time)**: desde que el cliente realiza una solicitud hasta que los miembros del equipo comienzan con el trabajo.
- **Tiempo de ciclo (cycle time)**: desde que el trabajo comienza hasta que el producto se entrega al cliente listo para su uso. Este tiempo de ciclo podría dividirse en el que agrega valor y en aquel que no agrega valor (ej. esperas, ineficiencias, etc.)
- **Tiempo de entrega (lead time)**: desde que el cliente realiza la solicitud hasta que finaliza la entrega del producto. O sea, la sumatoria del tiempo de respuesta más el tiempo de ciclo.

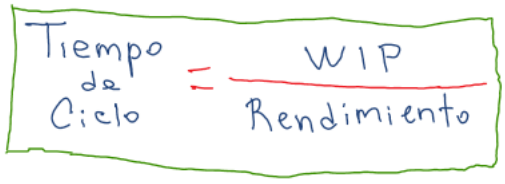


- **Tiempo ritmo (takt time)**: ritmo necesario para satisfacer la demanda del cliente. Por ejemplo, si el cliente quiere que se completen 30 puntos de historia en 15 días, el ritmo de producción promedio debería ser de 2 puntos de historia por día.

  Ritmo = Cantidad de productos que se quieren / Tiempo disponible

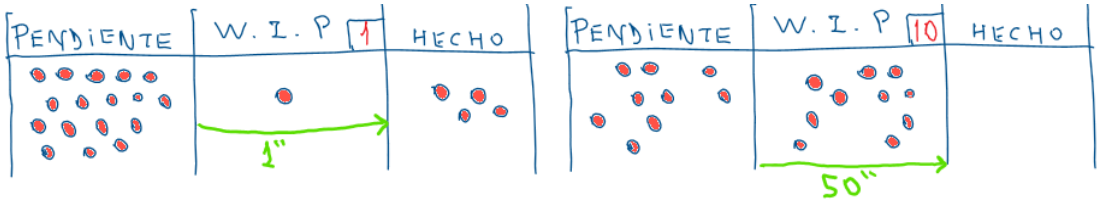
- **Rendimiento (throughput)**: capacidad promedio de producción de los miembros del equipo. Por ejemplo, completar 5 puntos de historia promedio por día.

John Little definió la relación entre el tiempo de ciclo, WIP y rendimiento:



O sea, si queremos acortar el tiempo de ciclo para entregar valor al cliente más rápido, deberíamos disminuir el WIP y/o aumentar el rendimiento.

Volvamos al ejemplo simple de que tenemos que guardar 10 pelotitas de ping pong en una caja. Si el WIP es limitado a un máximo de una pelotita, podríamos guardar 1 pelotita por segundo. Pero si nos lanzan 10 pelotitas juntas, se nos van a caer, tendremos que levantarlas y el tiempo promedio de producción podría elevarse a 5 segundos por pelotita, lo que es lo mismo a decir que el rendimiento bajaría a 0,2 pelotita por segundo.



Tiempo de ciclo ineficiente = 10 pelotitas / 0,2 pelotitas por segundo = 50 segundos

Tiempo de ciclo eficiente = 1 pelotita / 1 pelotita por segundo = 1 segundo

Por lo tanto, cuando el WIP es alto, para completar 10 pelotitas necesitamos un tiempo de ciclo de 50 segundos. Mientras que con un WIP bajo, podemos completar 1 pelotita por segundo (o 10 pelotitas en 10 segundos).

Una de las herramientas utilizadas para medir el tiempo de entrega y el tiempo de ciclo para varias tareas a través del tiempo es el diagrama de flujo acumulativo.

Diagrama de Flujo Acumulativo

El Diagrama de Flujo Acumulativo proporciona una visualización agregada a través del tiempo de métricas tales como:

Tiempos

- ✓ Respuesta
- ✓ Ciclo
- ✓ Entrega

Trabajo

- ✓ Solicitado
- ✓ En progreso
- ✓ Terminado

El principal objetivo del diagrama de flujo acumulativo es evaluar las variaciones del flujo del proyecto para anticipar cuellos de botella, detectar capacidad ociosa y corregir errores.

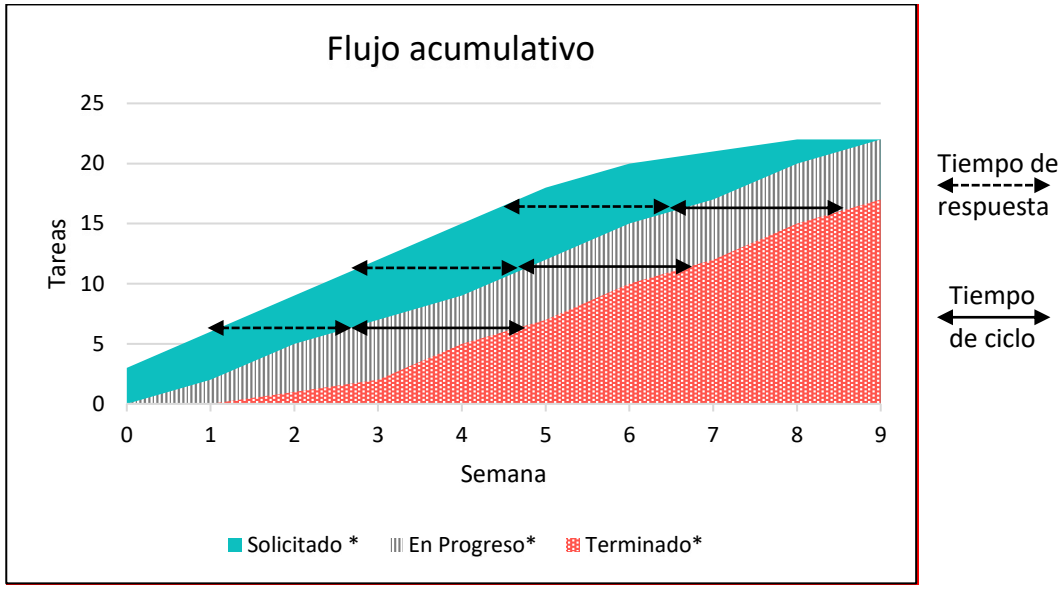
a) El progreso es estable

Veamos un ejemplo en la tabla a continuación donde se indica la cantidad de trabajo solicitado, en progreso y terminado durante 9 semanas de un proyecto.

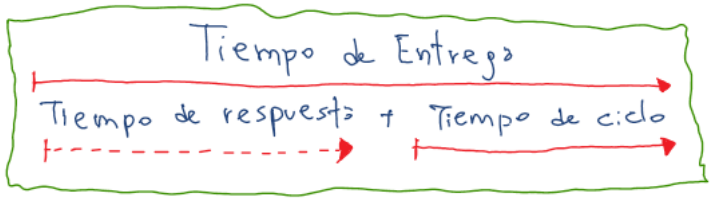
Semana	Trabajo Solicitado	Solicitado acumulado	Trabajo En progreso	En progreso acumulado	Trabajo Terminado	Terminado acumulado
0	3	3	0	0	0	0
1	3	6	2	2	0	0
2	3	9	3	5	1	1
3	3	12	2	7	1	2
4	3	15	2	9	3	5
5	3	18	3	12	2	7
6	2	20	3	15	3	10
7	1	21	2	17	2	12
8	1	22	3	20	3	15
9		22	2	22	2	17

Como se puede apreciar, el estado de este proyecto se mantiene relativamente estable a través del tiempo. Tanto el trabajo solicitado como el terminado semanalmente se mantienen entre 1 y 3 tareas, lo que permite mantener un trabajo en progreso constante entre 2 y 3 tareas.

Una forma más simple de analizar el estado del proyecto sería graficando el trabajo solicitado, trabajo en progreso y trabajo terminado en términos acumulativos a través del tiempo como se presenta en el diagrama a continuación.



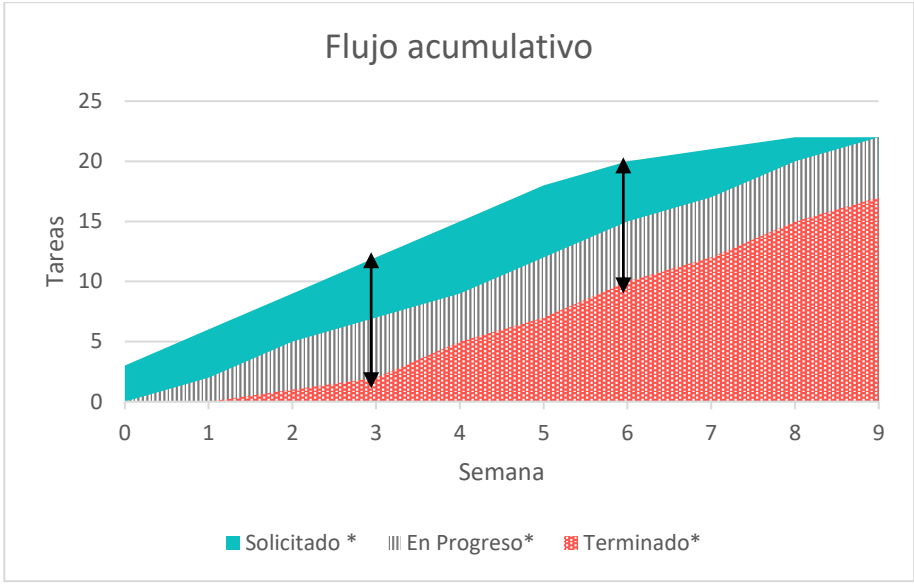
La diferencia horizontal del trabajo solicitado acumulado está indicando el tiempo de respuesta. Mientras que la diferencia horizontal del trabajo en progreso acumulado indica el tiempo de ciclo. La sumatoria del tiempo de respuesta más el tiempo de ciclo es el tiempo de entrega.



Si las bandas acumuladas se mantienen en paralelo, significa que el tiempo de respuesta y el tiempo de ciclo están estables. En este ejemplo, el tiempo de respuesta se ha mantenido en un valor aproximado a 2 semanas y el tiempo de ciclo también se ha mantenido estable en aproximadamente 2 semanas, por lo que el tiempo de entrega está estable en 4 semanas.

Esta estabilidad sería un escenario ideal, donde no vemos cuellos de botella o recursos ociosos. Por lo tanto, podríamos destinar esfuerzos en acortar los tiempos de respuesta y de ciclo de las tareas, eliminando los tiempos sin valor agregado.

Por otro lado, la diferencia vertical entre el trabajo solicitado acumulado menos el trabajo terminado acumulado, nos está indicando el trabajo pendiente



Por último, podemos calcular el trabajo en espera que todavía no ha comenzado:

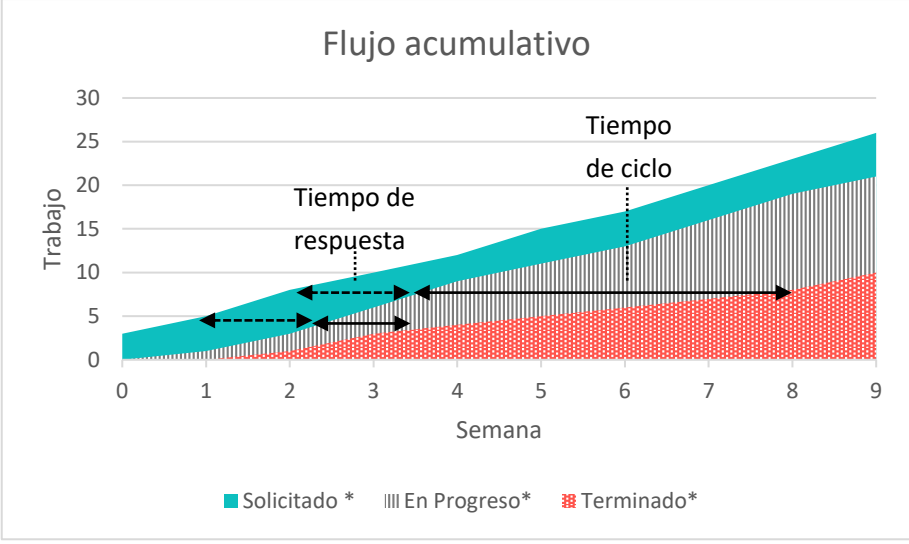
Trabajo en espera = Trabajo solicitado – Trabajo en progreso – Trabajo terminado

Semana	Solicitado acumulado	En progreso acumulado	Terminado acumulado	Trabajo en espera
0	4	0	0	4
1	7	2	0	5
2	10	5	1	4
3	13	7	2	4
4	16	9	5	2
5	19	12	7	0
6	21	15	10	0 *
7	22	17	12	0 *
8	23	20	15	0 *
9	23	22	17	0 *

* El trabajo en espera no puede ser negativo

b) Aumenta el tiempo de ciclo

Si en el diagrama de flujo acumulativo observamos que una de las bandas se está ensanchando, seguramente la cantidad de trabajo que ingresa en el tablero Kanban es mayor que el número de tareas que lo están dejando.



En el gráfico se puede observar que la banda de trabajo en progreso se ha ensanchado a través del tiempo, lo que significa que el tiempo de ciclo ha aumentado. Por su parte, la banda de trabajo solicitado acumulado se ha mantenido estable, por lo que el tiempo de respuesta se ha mantenido estable.

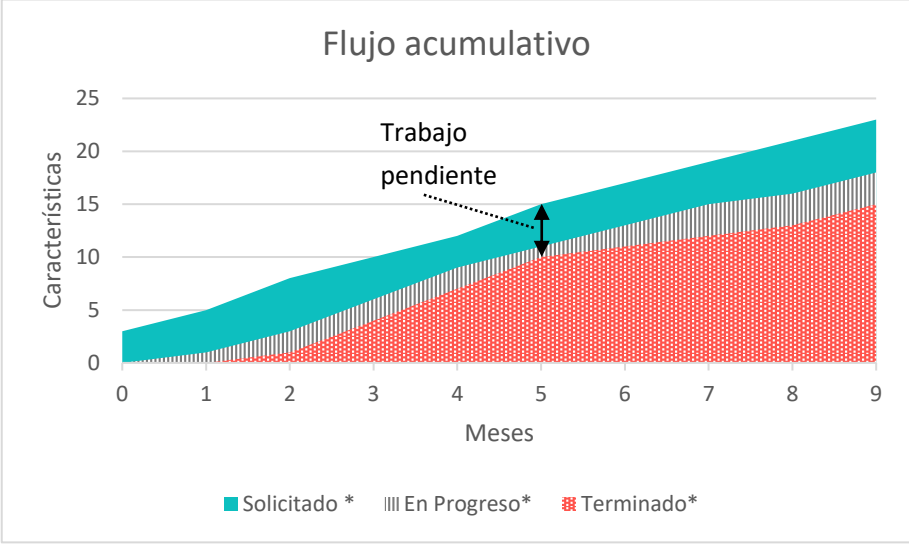
Si el tiempo de respuesta está estable, pero el tiempo de ciclo está aumentando, entonces el tiempo de entrega está aumentando, lo que es un problema para el cliente.

Si el tiempo de ciclo está aumentando, es probable que se haya generado un cuello de botella con un WIP más alto de lo recomendado. Esto suele ser un problema común causado por la multitarea y otras actividades de desperdicio que no generan valor, por lo que podríamos reconsiderar una disminución de los límites del WIP para concentrarnos más en finalizar las tareas en curso antes de comenzar con otras nuevas que nos generan cuellos de botella.

Es preferible tener pocas tareas terminadas en lugar de muchas tareas sin terminar

c) Capacidad ociosa

Si en el diagrama de flujo acumulativo, una banda se está reduciendo rápidamente, podría ocurrir que el rendimiento es mayor que la tasa de entrada de nuevas tareas, por lo que podríamos tener más capacidad de la necesaria en esa etapa y podríamos reubicar los recursos ociosos para optimizar el flujo.



En el gráfico se observa que en el mes 5, la banda del trabajo en progreso (WIP) se redujo rápidamente. Eso significa que podría existir más capacidad de recursos de lo necesario y podríamos aumentar el WIP para optimizar el flujo.

¿Cómo logramos reducir el tiempo de ciclo y entrega?

Podemos concluir que el tiempo de ciclo y entrega son estadísticas poderosas que proporcionan información valiosa sobre el flujo de trabajo, ayudan a identificar cuellos de botella y a comprender la capacidad de trabajo del equipo. Estos tiempos se pueden mejorar:

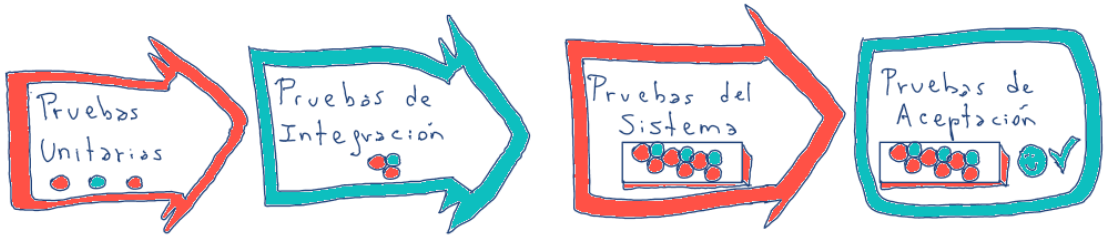
- Trabajando con solicitudes relativamente pequeñas para gestionarlas de manera ágil
- Reduciendo la cantidad de tareas que se encuentran en proceso (WIP)
- Mejorando el rendimiento (ej. reduciendo los errores y retrabajo)
- Reasignando los recursos según existan cuellos de botella o capacidad ociosa

2.8 Entregables de calidad

Antes de realizar la entrega del incremento del producto al cliente, es importante que el entregable haya sido sometido a un control de calidad. Para ello, en los proyectos ágiles se suelen aplicar distintas pruebas y prácticas como las que presentamos a continuación.

Pruebas en el desarrollo de software

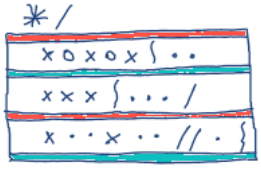
Por lo general, durante el proceso de desarrollo de software se suelen realizar como mínimo cuatro pruebas para asegurar la calidad del producto final.



1º Pruebas unitarias (Unit testing)

Una prueba unitaria es un fragmento breve del programa escrito que ejerce una parte estrecha del código fuente del producto y verifica los resultados.

En las pruebas unitarias el desarrollador prueba cada componente individual del producto.



El resultado de una prueba unitaria es "pasar (verde)" o “fallar (rojo)” según cuál sea la consistencia del comportamiento del programa con relación a las expectativas registradas.

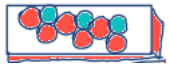
Los desarrolladores suelen escribir una gran cantidad de pruebas unitarias correspondientes a una gran cantidad de comportamientos de interés del programa, denominado "conjunto de pruebas".

Limitación: no se pueden aplicar las pruebas unitarias para aquellos sistemas que tienen mucha interdependencia entre los diferentes módulos.

2º Pruebas de integración (Integration testing)

Durante las pruebas de integración las unidades individuales se combinan y se prueban como un grupo. El propósito de este nivel de prueba es exponer fallas en las interfaces y en las interacciones entre los diferentes componentes o sistemas integrados.

También se utilizan estas pruebas para probar interfaces con organizaciones externas, por ejemplo, el intercambio electrónico de datos con servidores externos.

3º Pruebas del sistema (System testing) 

Prueba del sistema significa probar el sistema como un todo. Todos los módulos y componentes están integrados para verificar si el producto cumple con los requisitos especificados. Por ejemplo, verificar si el hardware y software cumplen con los requisitos.

Estas pruebas la realizan tanto los probadores como los desarrolladores.

4º Pruebas de aceptación (User Acceptance Testing)

Las pruebas de aceptación del usuario (UAT: User Acceptance Testing) consisten en probar el software por el usuario o cliente para determinar si es aceptado o no según su funcionalidad y rendimiento.

Estas son las últimas pruebas antes del lanzamiento y se realizan después de haber probado:

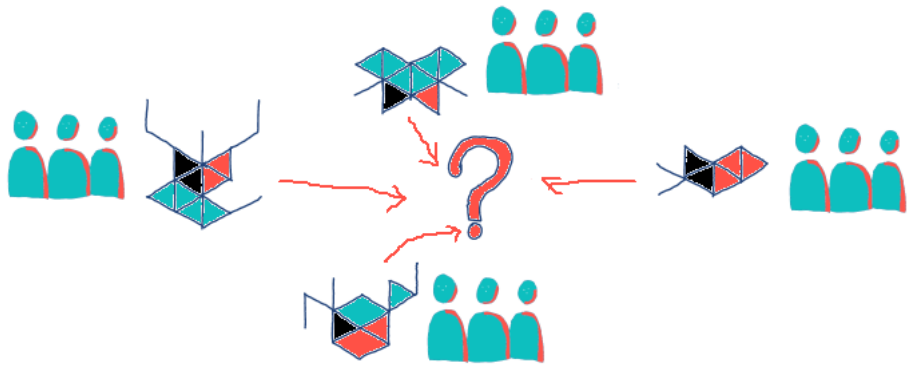
- ✓ Que cada módulo individual funcione correctamente
- ✓ Que todos los módulos se integran correctamente
- ✓ Que el software realice las funciones esperadas

Estas pruebas se diseñan a partir de las especificaciones de requisitos, los casos de uso y los procesos de negocio, y se enfocan en verificar si el sistema es “apto para el uso”.

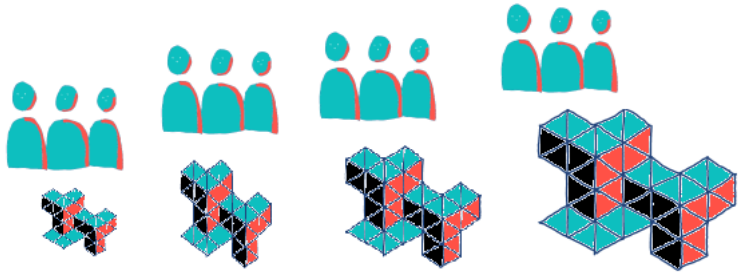
Las pruebas de las versiones “Beta” (Beta testing) con los usuarios reales de ese producto, forman parte de las pruebas de aceptación. Una versión Beta representa generalmente la primera versión completa de un programa, que es posible que sea inestable, pero útil para que sea considerada como una versión preliminar para probar en el mercado.

Integración continua sincrónica

La integración de un software plantea problemas cuando diferentes equipos diseñan y programan diferentes partes de una aplicación. Sobre el final del proyecto, cuando se intentan compilar todas las partes para terminar el producto, aparecen los problemas. Esto ocasiona en varias oportunidades una larga fase de desarrollo para corregir las fallas, unir los fragmentos de código, realizar las pruebas y lograr que el sistema funcione correctamente.



Para mitigar estos inconvenientes, la integración continua consiste en hacer integraciones automáticas (compilación y ejecución de pruebas) varias veces al día para detectar fallos cuanto antes. Por ejemplo, luego de cada hora de desarrollo se descargan las fuentes desde el control de versiones, se compilan, ejecutan y generan informes.



En la integración continua, los desarrolladores integran su código varias veces al día en el código fuente al que todos los programadores pueden acceder. Como se trabaja con pequeños fragmentos de código, la integración se realiza en pocos minutos. Si se descubre algún error en este proceso, se podrá localizar y solucionar rápidamente.

El objetivo de la integración continua es lograr un proceso de desarrollo más efectivo y poder reaccionar con más flexibilidad y rapidez ante los cambios.

Todos los involucrados en el desarrollo del software deberían poder acceder fácilmente al código fuente y ejecutar el archivo, por tal motivo, la integración continua exige que todos los archivos se encuentren en un único repositorio conocido por todos.

Ventajas

Los desarrolladores pueden detectar y solucionar problemas de integración de forma temprana y continua, evitando el caos de última hora

Disponibilidad constante de una versión para pruebas, demos o lanzamientos anticipados

Ejecución inmediata de las pruebas unitarias

Monitorización continua de las métricas de calidad del proyecto

Desventajas

Difícil de mantener. Por ejemplo, crear un repositorio de código automatizado implica que los equipos deben crear el conjunto de pruebas adecuado y dedicar tiempo a escribir casos de prueba en lugar de desarrollar código.

Puede derivar en demoras cuando varios desarrolladores intentan integrar sus códigos al mismo tiempo.

Desarrollo guiado por pruebas (TDD)

El desarrollo guiado por pruebas (TDD: Test-driven development) es una práctica de ingeniería de software que involucra dos prácticas:

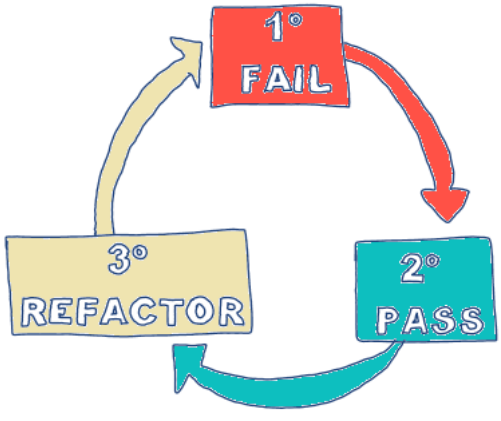
> **Escribir las pruebas primero** (Test First Development): seguir el ciclo "Escribir una prueba automatizada fallida -> Ejecutar la prueba fallida -> desarrollar código para hacer pasar la prueba -> ejecutar prueba -> repetir".
>
> Para escribir las pruebas generalmente se utilizan las pruebas unitarias (unit test) donde el desarrollador prueba cada componente individual del producto.
>
> **Refactorización** (Refactoring): eliminar códigos duplicados o innecesarios

El proceso resumido de TDD, también conocido como red-green-refactor, sería el siguiente:

1º Escribir una prueba unitaria y verificar que falle (Red)

2º Escribir el código que hace que la prueba pase satisfactoriamente (Green)

3º Refactorizar el código escrito para limpiar códigos duplicados o innecesarios (Refactor)

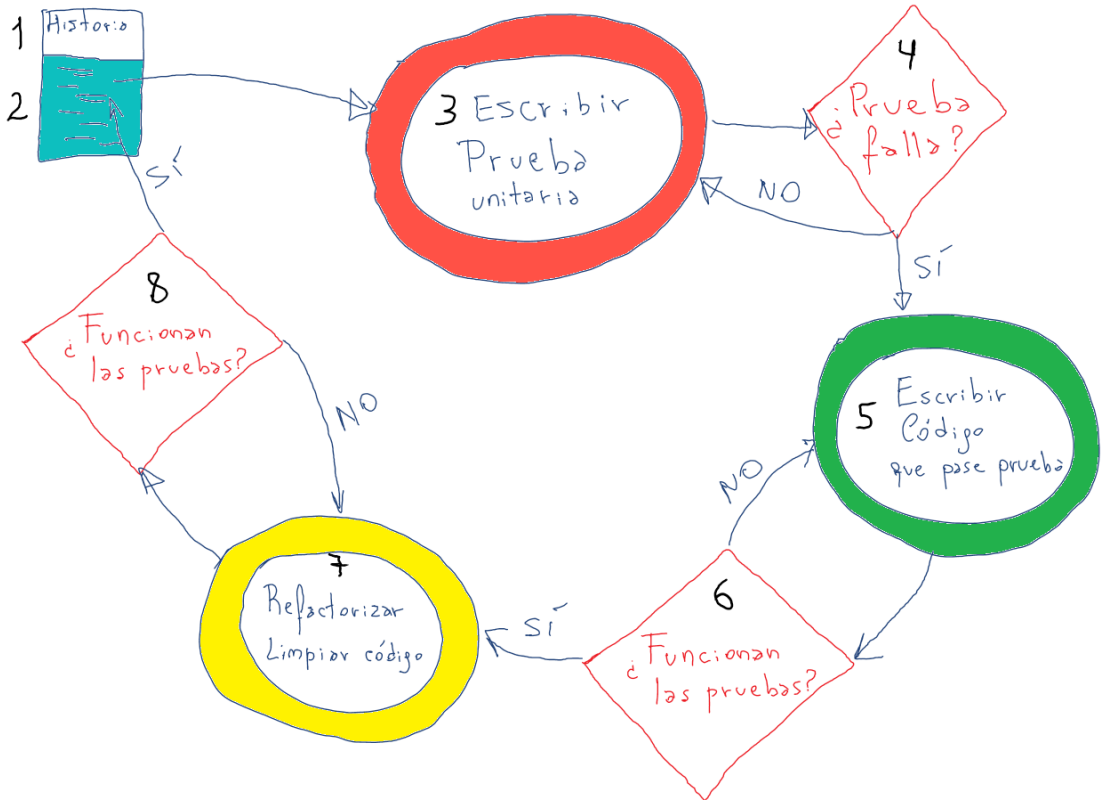


En TDD los requisitos se traducen en pruebas, de este modo, cuando las pruebas pasan se garantiza que el software cumple con los requisitos que se han establecido.

TDD desarrolla códigos limpios con mayor rapidez de desarrollo, más robustos, más seguros y fáciles de mantener.

Veamos un ejemplo más completo de como sería el proceso de TDD

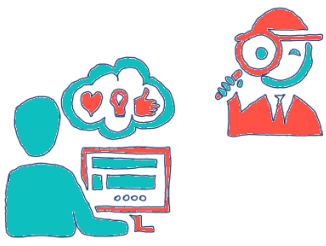
1. El cliente escribe su historia de usuario.
2. El equipo de proyecto escribe junto con el cliente los criterios de aceptación de esa historia, desglosándolos en pequeños criterios para simplificarlos la máximo posible.
3. Se escoge el criterio de aceptación más simple y se escribe en una prueba unitaria.
4. Se comprueba que la prueba falle.
5. Se escribe el código que hace pasar la prueba.
6. Se ejecutan todas las pruebas automatizadas para comprobar su funcionamiento
7. Se refactoriza y se limpia el código.
8. Se vuelven a pasar todas las pruebas automatizadas para comprobar que todo sigue funcionando.
9. Volvemos al punto 3 con los criterios de aceptación que falten y repetimos el ciclo una y otra vez hasta completar nuestra aplicación.



Profesional Ágil – Pablo Lledó ©

Pruebas de usabilidad

Cuando el producto o servicio está suficientemente completo, el probador (que imita al usuario real o es un usuario real) interactúa con el sistema para ver qué tan fácil es de usar y así poder descubrir potenciales problemas.



Durante la prueba de usabilidad los usuarios tratarán de completar tareas típicas del producto mientras los observadores escuchan y toman notas para identificar problemas, recopilar datos y determinar la satisfacción del participante con el producto.

El objetivo principal de esta prueba es verificar si los usuarios pueden usar el producto con facilidad y exponer los problemas de usabilidad que pueden necesitar cambios. Cuando se verifique que todo funcione correctamente, el producto estará listo para entregar al cliente.

Métricas

Si bien es importante realizar encuestas cualitativas sobre la experiencia y satisfacción del usuario sobre el producto, los indicadores cuantitativos que se mencionan a continuación sirven para verificar si lo que contestan los usuarios es lo que ocurre en la realidad.

- Porcentaje de usuarios que pudieron completar una tarea
- Tiempo que demora un usuario en completar una tarea
- Cantidad de errores al completar una tarea

Beneficios

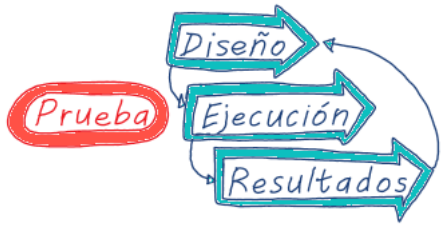
- Detectar y corregir problemas antes del lanzamiento
- Obtener información sobre los diferentes usos que podría tener el producto
- Descubrir los errores frecuentes de uso y las mejoras para facilitar el uso

Pruebas exploratorias

Las pruebas exploratorias también se denominan testing exploratorio (Exploratory testing).

A diferencia de las pruebas guionizadas (scripted testing) donde la prueba primero se diseña, luego se ejecuta y finalmente se interpreta si cumple o no con los resultados; en las pruebas exploratorias el resultado esperado puede ser más genérico o flexible y el probador (tester) tiene mayor libertad para ir aprendiendo de la prueba sobre la marcha y modificando los próximos pasos en base a lecciones aprendidas.

Con las pruebas exploratorias el diseño, la ejecución y aprendizaje sobre los resultados de las pruebas se realizan de forma simultánea.

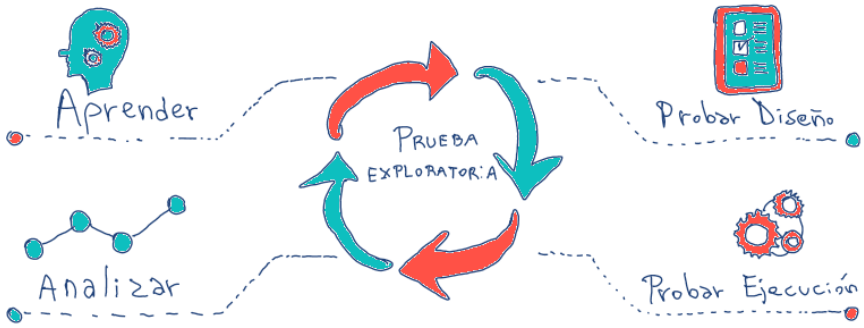


Los probadores aplican su autonomía, habilidades, pensamiento crítico y creatividad para buscar problemas, que sólo pueden ocurrir bajo circunstancias inusuales o imprevistas. Para ello, es necesario que piensen fuera de la caja para encontrar casos de prueba en tiempo real que les permita encontrar defectos.

El objetivo del testing exploratorio es que a medida que avanza la sesión de pruebas aumente la calidad de las pruebas con relación al objetivo que se busca. O sea, se va aprendiendo sobre la aplicación y se vuelca ese aprendizaje en el diseño y ejecución de las próximas pruebas.

Las pruebas exploratorias se suelen utilizar para:

- ✓ Aprender sobre la aplicación de manera rápida
- ✓ Buscar bugs complejos que no fueron detectados mediante pruebas automatizadas, testeo o verificación manual de funcionalidades
- ✓ Mejorar la documentación de la aplicación o los casos de prueba

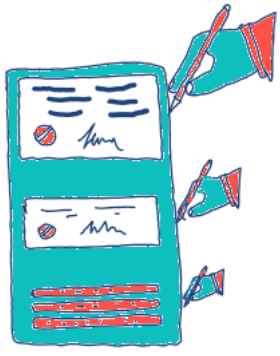


2.9 Contratos ágiles

En los proyectos que aplican metodologías ágiles, se suelen agregar cláusulas en los contratos con proveedores que incentiven la velocidad y que faciliten la flexibilidad por parte del cliente durante la ejecución del proyecto. A continuación, presentamos algunos ejemplos.

Contrato de varios niveles

Separar los elementos más cambiantes de un contrato de todo el resto, para permitir mayor flexibilidad y velocidad de contratación. Por ejemplo:



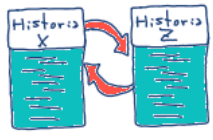
Nivel 1 - Los elementos comunes que no cambiarán como garantías, arbitraje, etc. se redactan en un acuerdo marco.

Nivel 2 - Los elementos sujetos a cambios (ej. tarifas, descripción del producto, etc.) se redactan en otro acuerdo.

Nivel 3 - Los elementos más dinámicos como el alcance, el cronograma y el presupuesto se formalizan en un acuerdo más liviano.

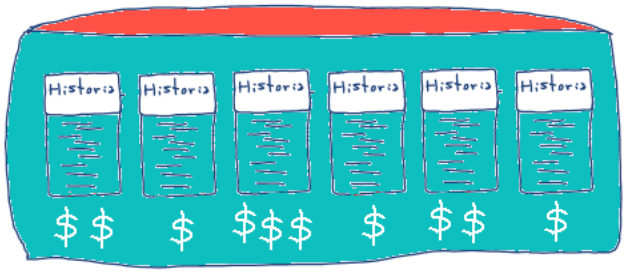
Alcance dinámico

El cliente puede variar el alcance del proyecto en puntos específicos acordados con el proveedor manteniendo un contrato de precio fijo. Por ejemplo, las horas de programadores que se iban a utilizar para hacer la historia X, se pueden reasignar para desarrollar la historia Z que requiere un esfuerzo similar.



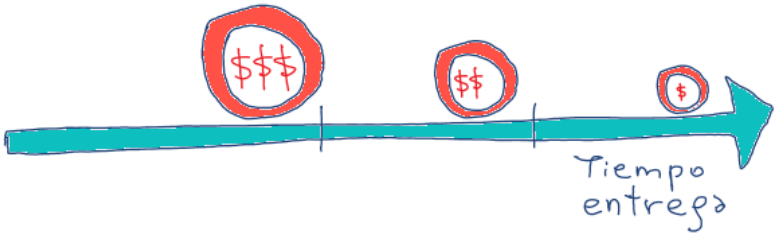
Incrementos de precio fijo

Descomponer el alcance en pequeños entregables (ej. historias de usuario) de precio fijo para que el cliente seleccione que entregable quiere que desarrollen sabiendo con anticipación el costo que deberá pagar por cada uno de ellos.



Tiempo y materiales graduados

El proveedor cobra una tarifa horaria más alta cuando entrega antes de lo acordado y cobra una tarifa menor en caso de demora en la entrega.



Cancelación anticipada

Cuando el proveedor entrega valor sin completar todo el alcance, el cliente no está obligado a pagar el trabajo pendiente, sino que puede pagar un cargo por cancelación anticipada. De esa forma ambas partes ganan, porque el cliente termina pagando menos que lo planificado y el proveedor recibe un pago por trabajo que no va a realizar.

2.10 Ejercicios – Valor

1 Marque con una X cuál de los siguientes ítems se incluyen en el producto viable mínimo.

Característica	Se incluye en MVP
Diseño para impresionar al usuario	
Diseño para obtener beneficios de manera inmediata	
Diseño para obtener una rápida retroalimentación	
Enfoque para resolver un problema	
Probar nuevas ideas para aprender	
Producto final	

2 Complete con las siglas MVP (Minimum viable product) o MMF (Minimum marketable producto), cada una de las siguientes frases.

A diferencia del M____ que podría ser un prototipo preliminar, M____ siempre es un producto terminado.

El M____ no puede estar basado solamente en las funcionalidades del producto, sino que debe incluir también algo de fiabilidad, usabilidad y diseño.

M____ es la unidad de funcionalidad más pequeña posible que proporciona un valor tangible a los clientes.

M____ prioriza las características de gran valor y menor tiempo de producción y comercialización, para lanzar productos más rápido al mercado y así aumentar el ROI.

3 Marque los criterios a considerar para estimar el valor de las historias de usuario.

Criterio	
Esfuerzo necesario o costos asociados a cada historia	
Generar un mayor retorno de la inversión	
Incrementar la participación de mercado	
Mejoras de la imagen empresarial	
Optimizar los procesos más importantes	
Reducir riesgos	
Resolver más problemas	
Resolver problemas tecnológicos para poder avanzar con el desarrollo del producto	

4 Coloque el tipo de ingreso (nuevo, incremental, retenido) que corresponde cada uno de los siguientes ejemplos.

Ejemplo	Ingreso
A los clientes actuales se les regala una funcionalidad incremental para que sigan comprando los servicios de mantenimiento de la empresa	
El incremento del producto comienza a venderse en otras regiones	
Los clientes actuales compran más unidades del producto	
Se agrega una nueva funcionalidad al producto cobrando un costo adicional a los clientes actuales	
Se desarrolla una nueva característica de acceso gratuito que evita que los clientes compren un producto de la competencia	
Una campaña de marketing aumenta las ventas del producto en consumidores mayores a 50 años que no eran clientes de la empresa	

5 Calcule el ROI de cada una de las siguientes alternativas.

Épica	Inversión	Mes 1	Mes 2	Mes 3	ROI
A	-$ 100	$ 25	$ 25	$ 55	
B	-$ 50	$ 30	$ 20	$ 10	
C	-$ 200	$ 0	$ 0	$ 250	

6 En la tabla a continuación se presenta el valor actual de 3 alternativas de inversión. Calcule el VAN de cada alternativa.

Épica	Inversión	Mes 1	Mes 2	Mes 3	VAN
A	-$ 100	$ 24	$ 23	$ 48	
B	-$ 50	$ 29	$ 18	$ 9	
C	-$ 200	$ 0	$ 0	$ 216	

7 Conteste verdadero (V) o falso (F).

	V o F
La TIR es la tasa de interés que hace el ROI igual a cero	
Si la TIR es 15% y el costo de oportunidad del dinero es 5%, el VAN es negativo	
Si la TIR es 3% y el costo del dinero es 3,5%, el VAN es positivo	
A menor TIR mayor prioridad debería tener la épica a desarrollar	

8 En base al método de los $100, calcule la prioridad de las siguientes historias de usuario.

Historia	A	B	C	D	E
María	20	20	20	20	20
José	40	40	20	0	0
Juan	0	10	20	30	40
Josefina	30	30	0	20	20
Prioridad					

9 Según el modelo de Kano, a qué tipo de calidad (requerida, unidimensional, entusiasmo, indiferente, inversa) corresponde cada uno de los siguientes ítems.

	Calidad
Aspectos que no son ni buenos ni malos; no generan satisfacción o insatisfacción del cliente.	
Dan como resultado la satisfacción cuando se cumplen e insatisfacción cuando no se cumplen porque fueron ofrecidos por el vendedor.	
No generan satisfacción cuando se cumplen porque son atributos básicos, pero dan lugar a gran insatisfacción cuando no se cumplen.	
Proporcionan satisfacción cuando se logran plenamente, pero no causan insatisfacción cuando no se logran porque no son esperados por el cliente ni fueron mencionados antes de la compra.	
Un alto grado de atributos resulta en la insatisfacción de algunos clientes.	

10 Según el model de priorización de MoSCoW, coloque la prioridad de cada una de las siguientes letras:

Letra	Prioridad
C	
M	
O	
S	
W	

11 Según el modelo puño-cinco, marque las opciones que bloquean una iniciativa.

Puño ___	1 ___	2 ___	3 ___	4 ___	5 ___

12 Según el modelo de priorización de Karl Weiger, ¿Cuál sería la prioridad de cada una de las siguientes historias de usuario?

Historia	Beneficio	Penalidad	Costo	Riesgo	**Prioridad**	
A	9	4	2	7		
B	2	8	7	3		
C	5	1	2	1		

13 Calcule la prioridad de las siguientes épicas ponderando en 80% el valor y en 20% el esfuerzo.

Épica	Valor (V)	Esfuerzo (E)	Prioridad
A	2	5	
B	8	5	
C	5	2	
D	13	13	

14 Complete con las palabras “fino” o “grueso” según corresponda.

Los elementos prioritarios del product backlog deben ser de grano _____, mientras que a medida que disminuye su prioridad, deberían volverse cada vez más de grano _____.

En las fases iniciales cuando el producto es joven, el product backlog suele ser de grano _____ relativo comparado con fases avanzadas donde el proyecto ya está más maduro y el grano es más _____.

15 El tiempo de desarrollo de una historia de usuario es de 80 horas. Hay otros tiempos relacionados con ese desarrollo como traslados (10 horas), esperas (30 horas) y almacenamiento (20 horas) que no agregan valor al cliente. ¿Cuál es el ratio de tiempo valor agrado?

Respuesta: _____

16 En la tabla a continuación complete al lado de cada acción de mitigación con alguna letra (TIMWOD) que se relacione con cada desperdicio.

Desperdicios del desarrollo de software		Cómo mitigarlo	
T	Trabajo parcialmente realizado o en proceso	Aprendizaje compartido	
I	Entregar características adicionales innecesarias	Colocar límites del WIP	
M	Reaprendizaje (aprender lo mismo)	Desarrollo basado en pruebas	
W	Traspasos o transferencias	Equipos autónomos y multifuncionales	
O	Retrasos	Reducir las dependencias entre los equipos al unir el trabajo relacionado	
D	Defectos	Validar supuestos sobre el valor en ciclos cortos de retroalimentación	

17 Complete con verdadero (V) o falso (F)

Cuando el WIP es _____	(V o F)
Alto podríamos tener miembros del equipo ociosos porque no podrán trabajar juntos todo el tiempo sobre esa misma tarea.	
Alto, los miembros del equipo podrían perder hasta el 20% de su tiempo si trabajaran con 2 tareas en simultáneo.	
Bajo fomenta la colaboración y trabajo en equipo.	
Demasiado bajo, existe el riesgo de empezar muchas tareas y terminar pocas.	
Muy alto, podría haber miembros del equipo ociosos sin estar trabajando en tareas.	
Muy alto, podría haber tareas en el listado del WIP pero que no tienen personas trabajando en ellas.	
Muy bajo, tendremos varias tareas en estado de “pendientes” demasiado tiempo, en espera de que el equipo comience a trabajar sobre ellas.	

18 Complete los tipos de tiempo (respuesta, ciclo, entrega, ritmo)

	Tipo de tiempo
Cantidad de productos que se quieren / Tiempo disponible	
Desde que el cliente realiza la solicitud hasta que finaliza la entrega del producto. O sea, la sumatoria del tiempo de respuesta más el tiempo de ciclo.	
Desde que el cliente realiza una solicitud hasta que los miembros del equipo comienzan con el trabajo.	
Desde que el trabajo comienza hasta que el producto se entrega al cliente listo para su uso. Este tiempo de ciclo podría dividirse en el que agrega valor y en aquel que no agrega valor	
WIP / Rendimiento	

19 Coloque el orden cronológico de las siguientes pruebas del desarrollo de software.

Pruebas	Orden cronológico
Aceptación	
Integración	
Sistema	
Unitarias	

20 Coloque a qué tipo de prueba (unitaria, integración, sistema, UAT, usabilidad, exploratoria) corresponde cada enunciado.

	Tipo de prueba
El diseño, la ejecución y aprendizaje sobre los resultados de las pruebas se realizan de forma simultánea.	
El probador interactúa con el sistema para ver qué tan fácil es de usar y así poder descubrir potenciales problemas.	
exponer fallas en las interfaces y en las interacciones entre los diferentes componentes	
Fragmento breve del programa escrito que ejerce una parte estrecha del código fuente del producto y verifica los resultados.	
las unidades individuales se combinan y se prueban como un grupo	
Pasar o fallar según cuál sea la consistencia del comportamiento del programa con relación a las expectativas registradas	
probar el software por cliente para determinar si es aceptado o no según su funcionalidad y rendimiento.	
Pruebas de las versiones Beta	
Todos los módulos y componentes están integrados para verificar si el producto cumple con los requisitos especificados	

21 Complete verdadero (V) o falso (F) en relación con la Integración Continua.

	V o F
Consiste en hacer integraciones automáticas (compilación y ejecución de pruebas) varias veces al día para detectar fallos cuanto antes.	
Los desarrolladores integran su código dos o tres veces por semana en el código fuente al que todos los programadores pueden acceder.	
La integración se realiza en un par de horas.	
El principal objetivo es lograr un proceso de desarrollo más efectivo y poder reaccionar con más flexibilidad y rapidez ante los cambios.	
Exige que todos los archivos se encuentren en un único repositorio conocido por todos.	
Los desarrolladores pueden detectar y solucionar problemas de integración de forma temprana y continua, evitando el caos de última hora.	
Disponibilidad constante de una versión para pruebas, demos o lanzamientos anticipados.	
Ejecución inmediata de las pruebas unitarias.	
Monitorización continua de las métricas de calidad del proyecto.	
Fácil de mantener.	
Puede derivar en demoras cuando varios desarrolladores intentan integrar sus códigos al mismo tiempo.	

22 Ordene cronológicamente los siguientes pasos del TDD

	Orden
Escribe en una prueba unitaria	
Repetimos el ciclo una y otra vez hasta completar nuestra aplicación	
Se comprueba que la prueba falle	
Se ejecutan todas las pruebas automatizadas para comprobar su funcionamiento	
Se escribe el código que hace pasar la prueba	
Se refactoriza y se limpia el código	
Se vuelven a ejecutar todas las pruebas para comprobar que todo funcione	

23 A qué tipo de contrato (varios niveles, alcance dinámico, incrementos de precio fijo, tiempo y materiales graduados, clausula de cancelación anticipada) corresponden las siguiente situaciones.

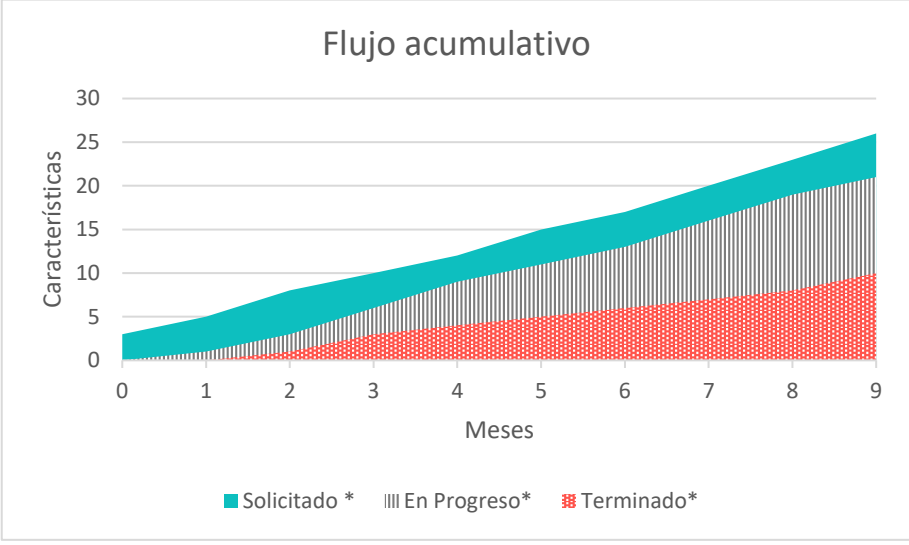
	Contrato
Separar los elementos más cambiantes de un contrato de todo el resto	
Los elementos dinámicos como el alcance, el cronograma y el presupuesto se formalizan en un acuerdo más liviano ligados a un contrato marco	
El cliente puede variar el alcance del proyecto en puntos específicos acordados con el proveedor manteniendo un contrato de precio fijo	
Las horas de programadores que se iban a utilizar para hacer la historia A, se pueden reasignar para desarrollar la historia C de esfuerzo similar	
El alcance se descompone en pequeñas historias de usuario de precio fijo	
El proveedor cobra una tarifa horaria más alta cuando entrega antes de lo acordado.	
El cliente paga una tarifa horaria más baja en caso de demora en la entrega.	
El cliente puede pagar un valor fijo y no queda obligado a pagar el trabajo pendiente de realizar	

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

24 El mes próximo realizarán el lanzamiento del producto viable mínimo que tendrá suficientes características para poder venderlo y proporcionar retroalimentación para las futuras versiones de ese producto. ¿Cuál de los siguientes ítems es MENOS probable que ocurra en este proyecto?

A. Minimizar riesgos con relación al producto con más funcionalidades
B. Ofrecer las funcionalidades mínimas viable a la población total de clientes
C. Retroalimentación temprana de los clientes
D. Minimizar el tiempo invertido en una iteración

25 Para el desarrollo de una moneda virtual que reemplazará todos los billetes físicos de un estado, se utiliza el método Kanban. Según el siguiente Diagrama de Flujo Acumulativo de las características completadas, ¿Cuál es la situación del proyecto al mes 8?



A. El tiempo de respuesta ha disminuido
B. El número de tarjetas que entran es menor que el número de tareas que salen
C. El tiempo de ciclo ha aumentado
D. El tiempo de entrega se ha mantenido estable

26 Para proteger los datos de ciudadanos de una ciudad digital se están definiendo épicas e historias de usuarios. ¿En cuál de los siguientes cuatro escenarios las historias de usuario del trabajo pendiente asociado al producto serán de grano relativamente más fino?

A. Proyecto en fases iniciales e historias de alta prioridad
B. Proyecto en fases avanzadas e historias de baja prioridad
C. Proyecto en fases iniciales e historias de baja prioridad
D. Proyecto en fases avanzadas e historias de alta prioridad

2.11 Respuesta ejercicios – Valor

Producto viable mínimo (MVP)

Característica	Se incluye en MVP
Diseño para impresionar al usuario	
Diseño para obtener beneficios de manera inmediata	
Diseño para obtener una rápida retroalimentación	√
Enfoque para resolver un problema	√
Probar nuevas ideas para aprender	√
Producto final	

MVP (Minimum viable product) vs MMF (Minimum marketable product)

A diferencia del MVP que podría ser un prototipo preliminar, MMF siempre es un producto terminado.

El MVP no puede estar basado solamente en las funcionalidades del producto, sino que debe incluir también algo de fiabilidad, usabilidad y diseño.

MMF es la unidad de funcionalidad más pequeña posible que proporciona un valor tangible a los clientes.

MMF prioriza las características de gran valor y menor tiempo de producción y comercialización, para lanzar productos más rápido al mercado y así aumentar el ROI.

3R Criterios para estimar el valor de las historias de usuario.

Criterio	
Esfuerzo necesario o costos asociados a cada historia	√
Generar un mayor retorno de la inversión	√
Incrementar la participación de mercado	√
Mejoras de la imagen empresarial	√
Optimizar los procesos más importantes	√
Reducir riesgos	√
Resolver más problemas	√
Resolver problemas tecnológicos para poder avanzar con el desarrollo del producto	√

4R Tipos de ingresos

Ejemplo	Ingreso
A los clientes actuales se les regala una funcionalidad incremental para que sigan comprando los servicios de mantenimiento de la empresa	Retenidos
El incremento del producto comienza a venderse en otras regiones	Nuevo
Los clientes actuales compran más unidades del producto	Incremental
Se agrega una nueva funcionalidad al producto cobrando un costo adicional a los clientes actuales	Incremental
Se desarrolla una nueva característica de acceso gratuito que evita que los clientes compren un producto de la competencia	Retenidos
Una campaña de marketing aumenta las ventas del producto en consumidores mayores a 50 años que no eran clientes de la empresa	Nuevo

5R ROI (Return on Investment)

Épica	Inversión	Mes 1	Mes 2	Mes 3	Ingresos	Ingreso neto	ROI
A	-$ 100	$ 25	$ 25	$ 55	$ 105	$ 5	$5 / $100 = 5%
B	-$ 50	$ 30	$ 20	$ 10	$60	$ 10	$10 / $50 = 20%
C	-$ 200	$ 0	$ 0	$ 250	$250	$ 50	$50 / $200 = 25%

6R VAN (Valor actual neto)

Épica	Inversión	Mes 1	Mes 2	Mes 3	Valor actual Ingresos	VAN (VA Ingresos – Inversión)
A	-$ 100	$ 24	$ 23	$ 48	$ 94	-$ 6
B	-$ 50	$ 29	$ 18	$ 9	$ 55	$ 5
C	-$ 200	$ 0	$ 0	$ 216	$ 216	$ 16

7R TIR (Tasa interna de retorno)

	V o F
La TIR es la tasa de interés que hace el ROI igual a cero Respuesta: TIR es la tasa de interés que hace el VAN igual a cero	Falso
Si la TIR es 15% y el costo de oportunidad del dinero es 5%, el VAN es negativo Respuesta: Si la TIR es mayor al costo del dinero, el VAN es positivo.	Falso
Si la TIR es 3% y el costo del dinero es 3,5%, el VAN es positivo Respuesta: Si la TIR es menor al costo del dinero, el VAN en negativo.	Falso
A menor TIR mayor prioridad debería tener la épica a desarrollar Respuesta: A mayor TIR, mayor prioridad.	Falso

8R Método de los 100 puntos o $100.

Historia	A	B	C	D	E	Total
María	20	20	20	20	20	100
José	40	40	20	0	0	100
Juan	0	10	20	30	40	100
Josefina	30	30	0	20	20	100
Total	$ 90	$ 100	$ 60	$ 70	$ 80	
Prioridad	2º	1º	5º	4º	3º	

9R Modelo de Kano

	Calidad
Aspectos que no son ni buenos ni malos; no generan satisfacción o insatisfacción del cliente.	Indiferente
Dan como resultado la satisfacción cuando se cumplen e insatisfacción cuando no se cumplen porque fueron ofrecidos por el vendedor.	Unidimensional
No generan satisfacción cuando se cumplen porque son atributos básicos, pero dan lugar a gran insatisfacción cuando no se cumplen.	Requerida
Proporcionan satisfacción cuando se logran plenamente, pero no causan insatisfacción cuando no se logran porque no son esperados por el cliente ni fueron mencionados antes de la compra.	Entusiasmo
Un alto grado de atributos resulta en la insatisfacción de algunos clientes.	Inversa

10R Según el model de priorización de MoSCoW, coloque la prioridad de cada una de las siguientes letras:

Letra	Prioridad
C	3º - Could have (podría implementarse)
M	1º - Must have (obligatorio)
O	No aplica
S	2º - Should have (deberían estar)
W	4 º - Won´t have (no lo queremos por el momento)

11R Modelo de votación puño-cinco.

Puño X	1 ___	2 ___	3 ___	4 ___	5 ___

Puño: bloqueará la propuesta porque cree que es perjudicial. A partir de un dedo no se bloqueará la iniciativa.

12R Modelo de priorización de Karl Weiger

Historia	Beneficio (a)	Penalidad (b)	**Valor (c)** (a) + (b)	Costo (d)	Riesgo (e)	**Costos (f)** (d) + (e)	**Prioridad** (c) / (f)	
A	9	4	13	2	7	9	**1,44**	2º
B	2	8	10	7	3	10	**1,00**	3º
C	5	1	6	2	1	3	**2,00**	1º

13R Priorización según valor y esfuerzo.

Épica	Valor (V)	80% V	Esfuerzo (E)	20% E	80% V / 20% E	Prioridad
A	2	1,6	5	1	1,60	**4º**
B	8	6,4	5	1	6,40	**2º**
C	5	4	2	0,4	10,00	**1º**
D	13	10,4	13	2,6	4,00	**3º**

14R Grano fino vs grano grueso

Los elementos prioritarios del product backlog deben ser de grano fino, mientras que a medida que disminuye su prioridad, deberían volverse cada vez más de grano grueso.

En las fases iniciales cuando el producto es joven, el product backlog suele ser de grano grueso relativo comparado con fases avanzadas donde el proyecto ya está más maduro y el grano es más fino.

15R Tiempo valor agregado

Desarrollo	80	valor agregado
Traslados	10	desperdicio
Esperas	30	desperdicio
Almacenamiento	20	desperdicio
Total	**140**	desperdicio
Ratio Valor agregado = 80 / 140		57,14%

16R TIMWOOD

Desperdicios del desarrollo de software		Cómo mitigarlo	
T	Trabajo parcialmente realizado o en proceso	Aprendizaje compartido	M
I	Entregar características adicionales innecesarias	Colocar límites del WIP	T
M	Reaprendizaje (aprender lo mismo)	Desarrollo basado en pruebas	D
W	Traspasos o transferencias	Equipos autónomos y multifuncionales	W
O	Retrasos	Reducir las dependencias entre los equipos al unir el trabajo relacionado	O
D	Defectos	Validar supuestos sobre el valor en ciclos cortos de retroalimentación	I

17R Work in Progress (WIP)

Cuando el WIP es _____	(V o F)
Alto podríamos tener miembros del equipo ociosos porque no podrán trabajar juntos todo el tiempo sobre esa misma tarea.	F
Alto, los miembros del equipo podrían perder hasta el 20% de su tiempo si trabajaran con 2 tareas en simultáneo.	V
Bajo fomenta la colaboración y trabajo en equipo.	V
Demasiado bajo, existe el riesgo de empezar muchas tareas y terminar pocas.	F
Muy alto, podría haber miembros del equipo ociosos sin estar trabajando en tareas.	F
Muy alto, podría haber tareas en el listado del WIP pero que no tienen personas trabajando en ellas.	V
Muy bajo, tendremos varias tareas en estado de “pendientes” demasiado tiempo, en espera de que el equipo comience a trabajar sobre ellas.	V

18R Tiempo: respuesta, ciclo, entrega, ritmo

	Tipo de tiempo
Cantidad de productos que se quieren / Tiempo disponible	ritmo
Desde que el cliente realiza la solicitud hasta que finaliza la entrega del producto. O sea, la sumatoria del tiempo de respuesta más el tiempo de ciclo.	entrega
Desde que el cliente realiza una solicitud hasta que los miembros del equipo comienzan con el trabajo.	respuesta
Desde que el trabajo comienza hasta que el producto se entrega al cliente listo para su uso. Este tiempo de ciclo podría dividirse en el que agrega valor y en aquel que no agrega valor	ciclo
WIP / Rendimiento	ciclo

19R Pruebas del desarrollo de software

Pruebas	Orden cronológico
Aceptación	4º
Integración	2º
Sistema	3º
Unitarias	1º

20R Tipos de pruebas

	Tipo de prueba
El diseño, la ejecución y aprendizaje sobre los resultados de las pruebas se realizan de forma simultánea.	Exploratoria
El probador interactúa con el sistema para ver qué tan fácil es de usar y así poder descubrir potenciales problemas.	Usabilidad
exponer fallas en las interfaces y en las interacciones entre los diferentes componentes	Integración
Fragmento breve del programa escrito que ejerce una parte estrecha del código fuente del producto y verifica los resultados.	Unitaria
las unidades individuales se combinan y se prueban como un grupo	Integración
Pasar o fallar según cuál sea la consistencia del comportamiento del programa con relación a las expectativas registradas	Unitaria
probar el software por cliente para determinar si es aceptado o no según su funcionalidad y rendimiento.	UAT
Pruebas de las versiones Beta	UAT
Todos los módulos y componentes están integrados para verificar si el producto cumple con los requisitos especificados	Sistema

21R Integración Continua.

	V o F
Consiste en hacer integraciones automáticas (compilación y ejecución de pruebas) varias veces al día para detectar fallos cuanto antes.	V
Los desarrolladores integran su código dos o tres veces por semana en el código fuente al que todos los programadores pueden acceder.	F
La integración se realiza en un par de horas.	F
El principal objetivo es lograr un proceso de desarrollo más efectivo y poder reaccionar con más flexibilidad y rapidez ante los cambios.	V
Exige que todos los archivos se encuentren en un único repositorio conocido por todos.	V
Los desarrolladores pueden detectar y solucionar problemas de integración de forma temprana y continua, evitando el caos de última hora.	V
Disponibilidad constante de una versión para pruebas, demos o lanzamientos anticipados.	V
Ejecución inmediata de las pruebas unitarias.	V
Monitorización continua de las métricas de calidad del proyecto.	V
Fácil de mantener.	F
Puede derivar en demoras cuando varios desarrolladores intentan integrar sus códigos al mismo tiempo.	V

22R Desarrollo guiado por pruebas (TDD)

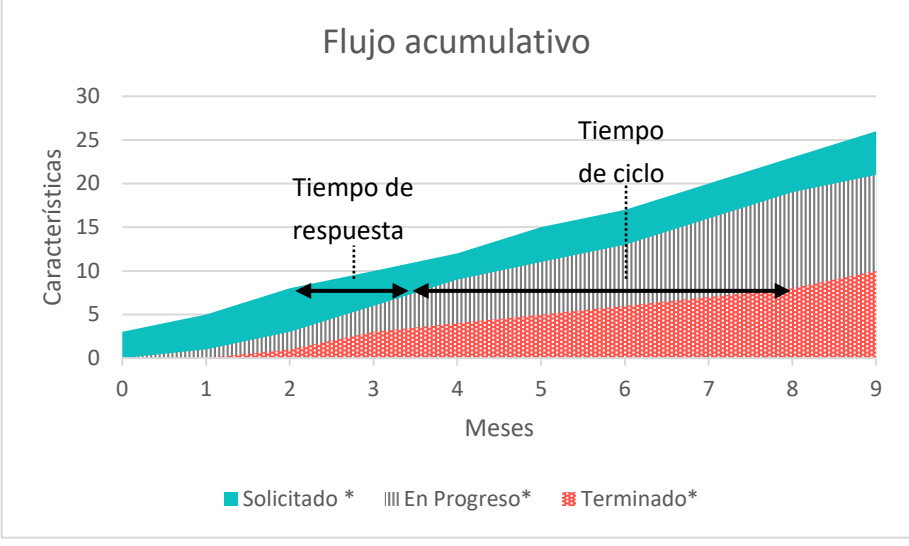
	Orden
Escribe en una prueba unitaria	1
Repetimos el ciclo una y otra vez hasta completar nuestra aplicación	7
Se comprueba que la prueba falle	2
Se ejecutan todas las pruebas automatizadas para comprobar su funcionamiento	4
Se escribe el código que hace pasar la prueba	3
Se refactoriza y se limpia el código	5
Se vuelven a ejecutar todas las pruebas para comprobar que todo funcione	6

23R Contratos ágiles

	Contrato
Separar los elementos más cambiantes de un contrato de todo el resto	Varios niveles
Los elementos dinámicos como el alcance, el cronograma y el presupuesto se formalizan en un acuerdo más liviano ligados a un contrato marco	Varios niveles
El cliente puede variar el alcance del proyecto en puntos específicos acordados con el proveedor manteniendo un contrato de precio fijo	Alcance dinámico
Las horas de programadores que se iban a utilizar para hacer la historia A, se pueden reasignar para desarrollar la historia C de esfuerzo similar	Alcance dinámico
El alcance se descompone en pequeñas historias de usuario de precio fijo	Incrementos precio fijo
El proveedor cobra una tarifa horaria más alta cuando entrega antes de lo acordado.	tiempo y materiales graduados
El cliente paga una tarifa horaria más baja en caso de demora en la entrega.	tiempo y materiales graduados
El cliente puede pagar un valor fijo y no queda obligado a pagar el trabajo pendiente de realizar	cancelación anticipada

24R Respuesta B. El producto viable mínimo (MVP) es la versión preliminar de un nuevo producto que permite al equipo obtener retroalimentación de los clientes para descubrir si están construyendo el producto correcto. / Aprender de un MVP es más barato que desarrollar un producto con más características, las cuales aumentan costes y riesgo si el producto falla. / Los desarrolladores típicamente lanzan el MVP para un subconjunto de los posibles clientes, como los "primeros seguidores", que son más tolerantes, más propensos a dar retroalimentación y capaces de captar la visión del producto a partir de un prototipo temprano. O sea, no lanzan el MVP a toda la población de clientes. / Con un MVP se busca minimizar el tiempo total gastado en una iteración. El proceso se itera hasta que se obtiene un producto que se ajusta al mercado, o se llega a la conclusión del que el producto no es viable.

25R Respuesta C. En el gráfico se puede observar que la banda de trabajo en progreso se ha ensanchado al mes 8. Esto significa que el número de tarjetas (solicitudes) que entran en la etapa correspondiente en el tablero Kanban es mayor que el número de tareas que lo están dejando. Por lo tanto, hay que reconsiderar los límites del WIP existentes en el tablero Kanban y concentrarse más en finalizar las tareas que están en curso antes de comenzar otras nuevas. / El tiempo de respuesta es la diferencia horizontal entre el trabajo en progreso acumulado y el trabajo solicitado acumulado. En este ejemplo, la banda de trabajo solicitado acumulado se ha mantenido estable, por lo que el tiempo de respuesta se ha mantenido estable. / El tiempo de ciclo es la diferencia horizontal entre el trabajo terminado acumulado y el trabajo en progreso acumulado. En el gráfico el tiempo de ciclo ha aumentado, lo que indica un problema con el WIP. / El tiempo de entrega se da por la sumatoria del tiempo de respuesta más el tiempo de ciclo. / Si el tiempo de ciclo está aumentando y el tiempo de respuesta está estable, entonces el tiempo de entrega también está aumentando.



26R Respuesta D. Los elementos prioritarios del product backlog deben ser de grano fino, mientras que a medida que disminuye su prioridad, deberían volverse cada vez más de grano grueso a nivel de épicas con historias de usuario con información incompleta. / Por otro lado, en las fases iniciales cuando el producto es joven, el product backlog suele tener menor nivel de detalle relativo comparado con fases avanzadas donde el proyecto ya está más maduro.

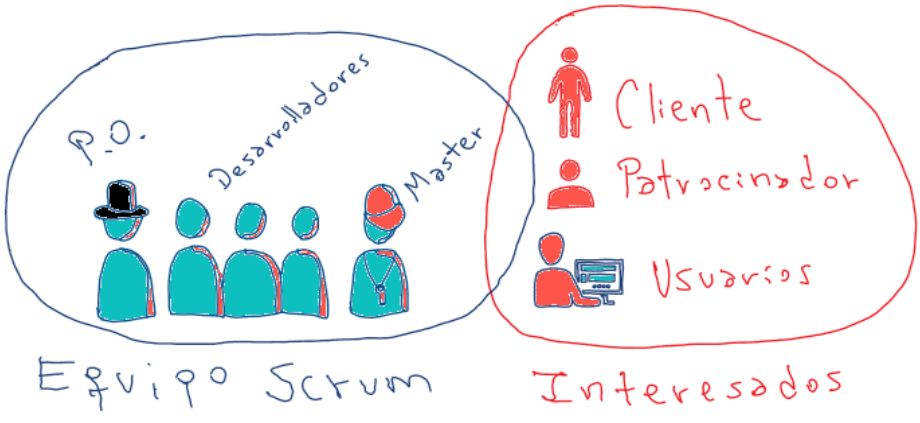
Capítulo 3

Involucrar a los interesados

3.1 Interesados

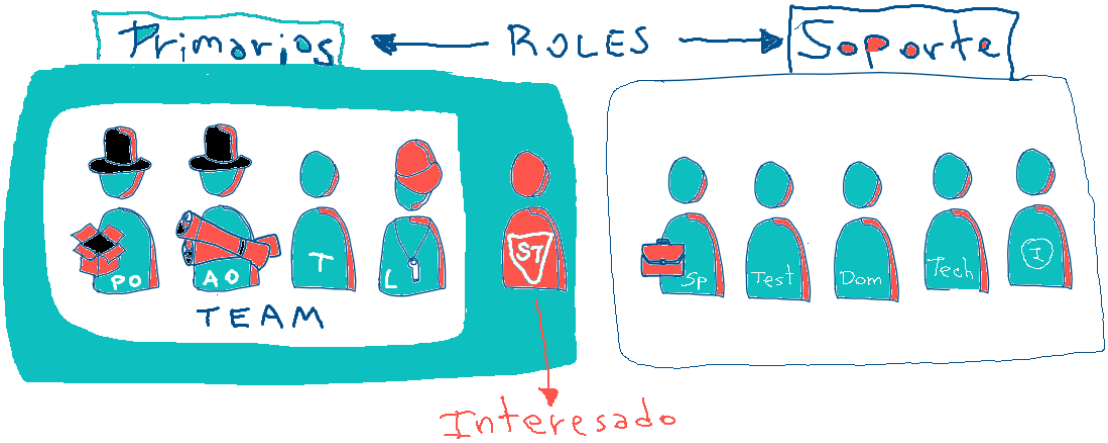
En la gestión tradicional de proyectos con ciclos predictivos, se entiende por interesados a todas aquellas personas u organizaciones que se verán afectadas por el proyecto. Por ejemplo: patrocinador, director del proyecto, miembros del equipo, proveedores, clientes, comunidad, etc. Sin embargo, en el ambiente de proyectos ágiles los miembros del equipo no forman parte de la definición de interesados.

Por ejemplo, en Scrum un interesado es cualquier persona con interés o influencia sobre el producto que no forma parte del equipo Scrum (dueño del producto, scrum master, equipo de desarrollo). Ejemplo de interesados en Scrum sería el cliente, el patrocinador y los usuarios.



El dueño del producto suele ser quien representa a los interesados en el equipo Scrum.

Por su parte, en Disciplined Agile los “interesados” no forman parte del equipo principal (dueño del producto, dueño de arquitectura, equipo de desarrollo, líder del equipo), ni del equipo de soporte (especialista, probador, experto de dominio, experto técnico, integrador).



Los interesados son personas u organizaciones que frecuentemente interactúan con el equipo para facilitar la creación de los productos influyendo durante todo el desarrollo del proyecto.

Para involucrar a los interesados en el proyecto de manera exitosa, será fundamental la colaboración y la comunicación clara y continua.

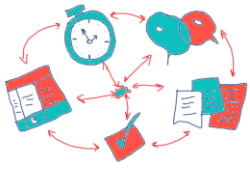
Procesos de gestión de los interesados

1º Identifique periódicamente a los interesados, en especial a los interesados clave que tendrán un gran impacto en el éxito del proyecto.

2º Conocer las necesidades de los interesados y los requisitos del proyecto mediante una estrecha comunicación

3º Involucrar a los interesados en el proyecto mediante la comunicación activa y el intercambio de la información apropiada para cada interesado.

4º Comunicar periódicamente el estado de avance del proyecto. Discutir estimaciones y proyecciones actualizadas de manera oportuna y abierta para facilitar la planificación futura.



5º Gestionar activamente las expectativas de los interesados a medida que avanza el proyecto, manteniendo una buena relación mediante la difusión de la información necesaria y la recopilación de la retroalimentación de los interesados.

Lo más importante de involucrar a los interesados a lo largo de todo el ciclo de vida del proyecto es detectar riesgos (negativos y positivos) para la mejora continua.

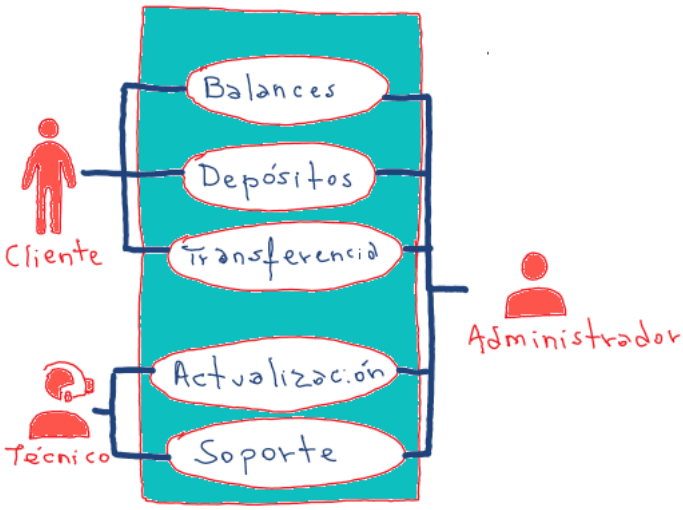
Diagrama del caso de uso

Los usuarios suelen formar parte de los interesados clave del proyecto, por lo tanto, es fundamental involucrarlos activamente en el proyecto.

Una herramienta para visualizar los usuarios de un sistema es el diagrama del caso de uso. Los tres elementos de este diagrama son:

- los usuarios
- los casos de uso
- las interrelaciones

Ejemplo de diagrama de caso de uso



El caso de uso es la descripción escrita de una funcionalidad del sistema que puede contener varias historias de usuario y puede requerir de varias iteraciones. Describe desde el punto de vista de un usuario, el comportamiento de un sistema cuando responde a una solicitud.

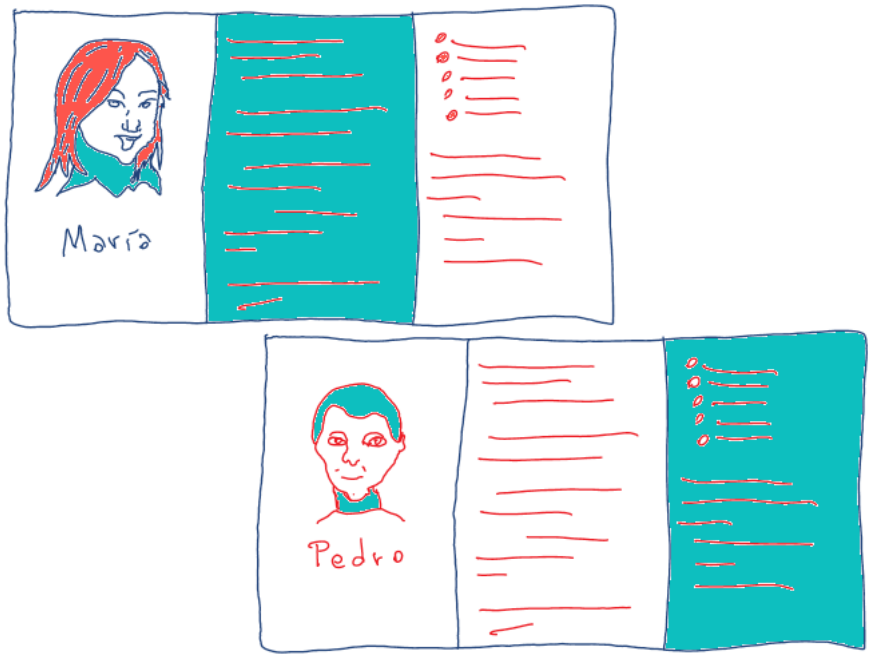
El diagrama del caso de uso describe cómo un usuario usa un sistema para lograr un objetivo particular.

Personas

Si bien los usuarios del sistema son interesados claves del proyecto, hay veces que no se tiene un acceso fácil a ellos para poder realizar el desarrollo y pruebas del producto antes de su lanzamiento. En estos casos se puede trabajar con el concepto de “persona”.

Una “persona” representa a un individuo ficticio, que se basa en usuarios reales, que interactúa con el sistema que se está desarrollando.

Cada persona se describe en una ficha con su nombre real, una imagen de rostro, su personalidad, motivaciones, edad, hobbies, experiencia laboral, familiares, etc.



Para describir personas de manera efectiva, hay que investigar y comprender a los usuarios reales. Las personas ayudan a guiar las decisiones sobre la funcionalidad y diseño del producto.

No deberíamos "inventar" personas, sino que la persona se descubre como resultado de una investigación para entender cuáles son las necesidades reales del usuario.

La persona principal debe quedar satisfecha con el producto, pero no puede estar satisfecho con una funcionalidad diseñada para otra persona.

Algunas acciones para involucrar a los interesados

1. Fomentar la participación temprana

Involucrar a los interesados en las fases iniciales del desarrollo del producto es fundamental para comprometerlos con el proyecto. Por ejemplo, invitar a los interesados clave en las discusiones preliminares sobre los requisitos o la definición de las historias de usuario.

2. Explicar los beneficios globales del producto

Explicar todos los beneficios que obtendremos con el producto aumenta el sentido de pertenencia sobre los impactos positivos del proyecto. Por ejemplo, no sólo explicar los beneficios directos que obtendrá la organización o los usuarios, sino los beneficios indirectos que obtendrá la comunidad al resolver un problema, mejorar su calidad de vida, etc.

3. Incluirlos en las prioridades del proyecto

Si bien el dueño del producto es el responsable final de establecer las prioridades de las historias de usuario a desarrollar, es fundamental mantener una comunicación fluida con los interesados para que ellos provean de información antes de decidir sobre las prioridades.

4. Involucrarlos en la planificación del lanzamiento

Si bien los miembros del equipo de desarrollo son los principales responsables de planificar las tareas necesarias a desarrollar durante las iteraciones antes de un lanzamiento, participar a los interesados será fundamental para aportar información sobre dependencias, riesgos, oportunidades, recursos compartidos, valor, etc.

5. Solicitar comentarios durante las revisiones de la iteración

Invitar a los interesados durante las reuniones de revisión de la iteración o revisión del producto para escuchar su retroalimentación es una excelente práctica para mantenerlos involucrados con el proyecto y mejorar la calidad.

3.2 Visión compartida

Para lograr un involucramiento activo de los interesados es necesario que los miembros del equipo y principales interesados compartan la visión del proyecto.

Algunas herramientas para lograr este objetivo suelen ser: acta de constitución, caja de visión, discurso del ascensor, hoja de ruta del producto, product backlog, definición de terminado, esquema de página, etc.

Acta de constitución del proyecto

El acta de constitución (o carta del proyecto) de un proyecto ágil es un resumen de alto nivel elaborado por el equipo que incluye la visión, misión, factores clave de éxito y los acuerdos entre el equipo y los interesados externos. Este acta es imprescindible para ayudar a crear una comprensión común del proyecto entre los interesados.

Suele tener el tamaño máximo de una hoja de rotafolio para que esté visible en la sala del equipo si están colocados, o una hoja en formato electrónico si trabajan de manera remota.

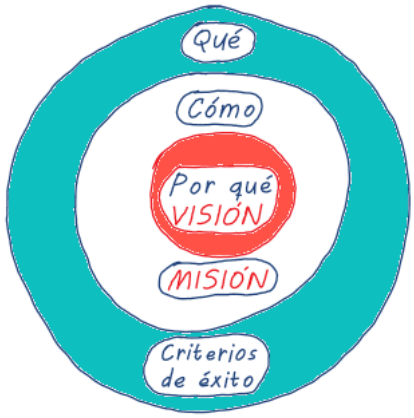
Este documento se puede utilizar para iniciar el proyecto ágil formalmente.

No se incluye información adicional como hitos, supuestos y restricciones que suelen formar parte del acta de constitución de proyectos de ciclos predictivos.

El acta de constitución ágil se enfoca más sobre "Cómo" se resolverá un problema o se gestionará el proyecto en lugar de "Qué" se desarrollará exactamente para resolver ese problema. De esta forma no se imponen límites innecesarios al alcance del proyecto.

Al principio del proyecto sabemos poco del producto o servicio final, por lo tanto, tendremos una versión resumida del acta que incluye solamente:

- Visión - ¿Por qué existe el proyecto?
- Misión - ¿Cómo va a trabajar el equipo para lograr la visión?
- Criterios de éxito - ¿Qué se logrará con el proyecto?



Cuando el proyecto está más avanzado, se podría agregar en este documento de una sola página información adicional como:

- Antecedentes
- Objetivos
- Interesados

Asegurar que los contenidos del acta de constitución sean conocidos y aprobados por todos los miembros del equipo será fundamental para que se involucren con el proyecto.

La visualización del proyecto podría cambiar a medida que se recopila nueva información, por lo que el acta de constitución deberá revisarse periódicamente para garantizar una comprensión clara de los interesados.

Caja de visión del producto

La caja de visión del producto (o arquitectura del proyecto) es una herramienta para simplificar la visualización del producto condensando la información en un espacio limitado.

Para realizar la caja de visión se suele realizar un taller práctico colaborativo con los miembros del equipo e interesados clave antes de comenzar con un nuevo proyecto. En este taller se obtendrá una visión compartida del producto que servirá para aceptar y rechazar los requisitos del proyecto.

Esta información resumida podría presentarse en una sola página o en una caja con el tamaño y formato de las utilizadas para cereales.

Ejemplo de visión del producto en una hoja

Visión del producto: _____			
Grupo objetivo	Necesidad	Producto	Objetivos
 ¿Quiénes serán beneficiados (clientes)? ¿Quién usarán el producto (usuarios)?	 ¿Por qué el cliente o usuario necesita el producto?	 ¿Qué características o funcionalidades son críticas para el cliente? ¿Qué hace que este producto sea diferente a los otros?	 ¿Cómo beneficiará el producto a la empresa que lo crea?

Ejemplo de caja de visión

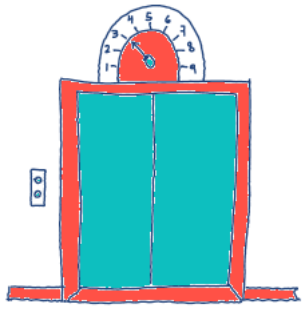
Frente de la caja
- ✓ Nombre del proyecto
- ✓ Logotipo y slogan
- ✓ Argumentos principales para comprar el producto

Laterales de la caja
- ✓ Necesidades técnicas
- ✓ Usuarios principales
- ✓ Funcionalidades principales

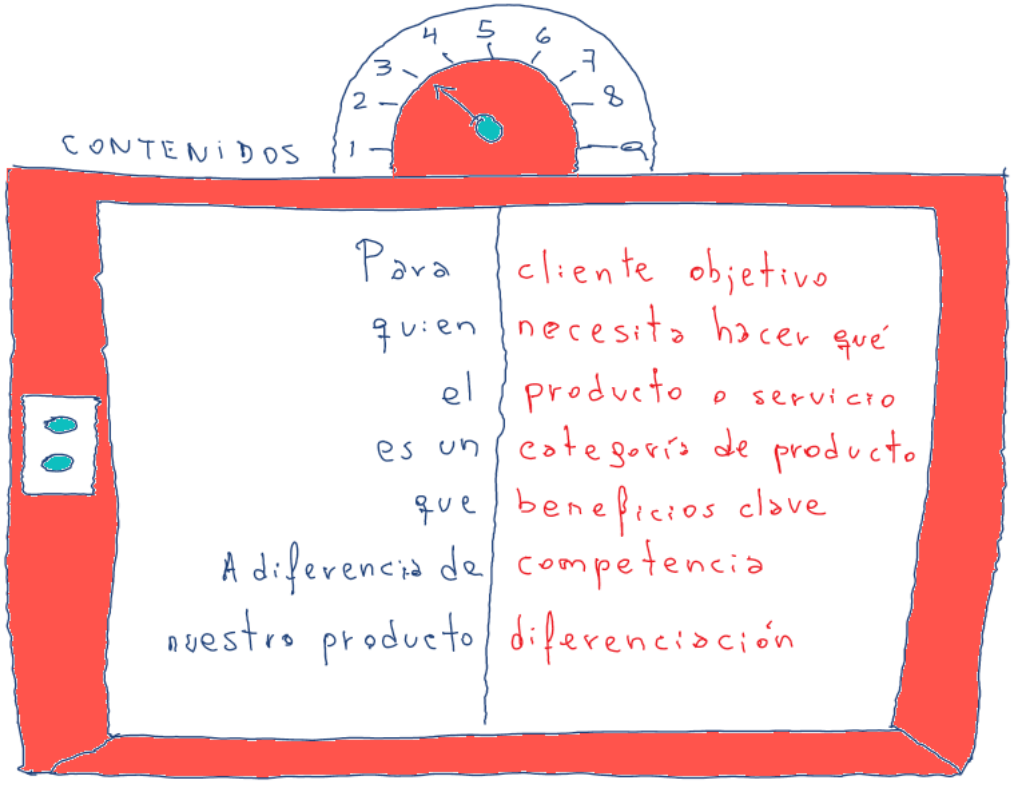
Parte posterior de la caja
- ✓ Características
- ✓ Funcionalidades avanzadas
- ✓ Requisitos del proyecto

Discurso del ascensor

El discurso del ascensor (elevator pitch o elevator statement) es una forma de articular la visión del producto para explicarla a una persona externa al proyecto en pocos minutos. Esta herramienta se utiliza para explorar, desarrollar y compartir una visión del producto.



Se denomina "discurso del ascensor" porque debe explicar brevemente la visión del producto a un interesado en el tiempo máximo que demora desde que sube a un ascensor hasta que se baje del mismo.



Ejemplo:

Para los gerentes financieros, **quienes** necesitan hacer un seguimiento de las operaciones de manera remota, **el** software "Remoting" **es un** sistema eficiente **que** realiza un seguimiento de las operaciones con reportes automatizados. **A diferencia de** los software actuales, **nuestro producto** se puede acceder de manera remota desde cualquier lugar en cualquier momento e incluye reportes gráficos que no tienen los de la competencia.

Hoja de ruta del producto

La hoja de ruta del producto (product roadmap) es una herramienta para describir cómo podría ser la evolución de un producto a través del tiempo. Por lo general, se estructura en términos de épicas y temas para reflejar la visión de largo plazo del proyecto.

Ejemplo de hoja de ruta

Fecha	Mes 3	Mes 6	Mes 9
Nombre (Lanzamiento o Épica)			
Objetivo	¿Por qué se va a desarrollar X?	¿Por qué se va a desarrollar Y?	¿Por qué se va a desarrollar Z?
Funcionalidades 	Principales funcionalidades de X 	Principales funcionalidades de Y 	Principales funcionalidades de Z
Métrica (Verificar objetivo) 	¿Cómo sabemos que se cumplió objetivo de X? 	¿Cómo sabemos que se cumplió el objetivo de Y? 	¿Cómo sabemos que se cumplió el objetivo de Z?

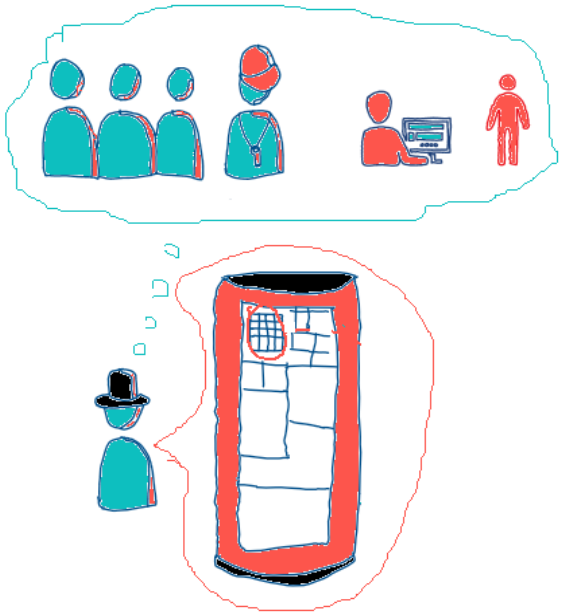
La hoja de ruta es utilizada para alinear a las partes interesadas y para adquirir un presupuesto preliminar para desarrollar el producto.

Esta herramienta debe ser un documento vivo en lugar de un plan inflexible. Debe discutirse, priorizarse, estimarse, actualizarse y compartirse con los interesados regularmente. Por ejemplo, si reduce la variable “recursos disponibles”, lo primero que deberíamos evaluar es un cambio del alcance para completar el proyecto en el tiempo disponible. En otras palabras, deberíamos reformular el alcance del lanzamiento con los interesados y volver a priorizar las historias de usuario pendientes para evaluar cuáles serían las características comercializables mínimas del nuevo producto para lanzar al mercado en el plazo establecido.

Trabajo pendiente asociado al producto (product backlog)

El listado de trabajo pendiente asociado al producto (product backlog) es un inventario ordenado por prioridades con las características del producto, generalmente compuesto de épicas o historias de usuario.

Si bien el responsable final de establecer las prioridades del product backlog es el dueño del producto que representa los intereses del cliente, es fundamental involucrar al equipo y a los interesados claves para que den su opinión sobre el producto a desarrollar. De esta forma, el dueño del producto decide sobre las prioridades teniendo una visión holística del producto.



El product backlog está en constante evolución y permanece disponible para que el equipo y los interesados contribuyan con sugerencias.

El equipo de desarrollo y el dueño del producto realizan un refinamiento continuo del listado de trabajo pendiente agregando detalles y estimaciones sobre las características a desarrollar.

En las fases iniciales del proyecto solamente se conoce el objetivo que se quiere resolver. Las características funcionales o historias de usuario que definan los interesados durante la iniciación podrán ir cambiando a lo largo de cada iteración del proyecto, por lo que se podría terminar con un producto diferente a lo que se visualizó en las primeras iteraciones.

Definición de Terminado (DoD)

La Definición de Terminado (DoD: Definition of Done) consiste en incluir todos los criterios requeridos para que un producto o lanzamiento sea considerado como 100% terminado para usar por el cliente.

Para evitar malentendidos sobre lo que significa un producto terminado, es importante que el equipo y los interesados clave estén de acuerdo en la definición de "terminado".

Ejemplos de DoD:

- ✓ Todos los elementos del código tienen que estar completados
- ✓ El código cumple con los estándares de desarrollo
- ✓ Pruebas unitarias escritas y aprobadas
- ✓ Los cambios de configuración fueron documentados y comunicados

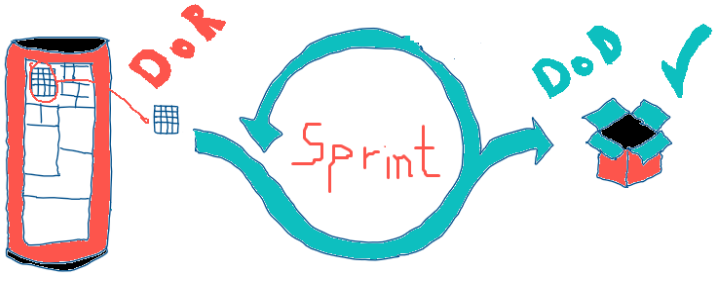
La definición de terminado alineará las expectativas de los interesados y el equipo del proyecto para reducir el riesgo de trabajar en algo que no es valorado por el cliente.

La definición de terminado podría ser para una funcionalidad, una iteración o un lanzamiento.

No hay que confundir la definición de terminado con la **definición de listo (DoR: Definition of Ready)** donde las historias de usuario tienen toda la información requerida por el equipo antes de comenzar a trabajar con ellas en una iteración.

Ejemplos de DoR:

- ✓ Historia de usuario clara, comprobable y factible
- ✓ Criterios de aceptación y dependencias de la historia de usuario definidos
- ✓ Persona que aceptará la historia de usuario identificada



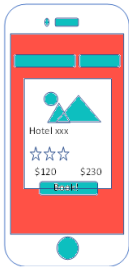
Wireframe, Mockup, Prototipo

Los wireframe, mockups y prototipos son excelentes herramientas para involucrar a los interesados con la definición del alcance del producto antes de incurrir en grandes inversiones para desarrollar el producto final.

Wireframe o Esquema de 1 página

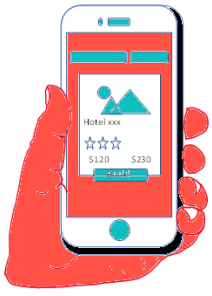
Esqueleto o estructura visual de un producto (ej. página web, app, o productos basados en pantallas).

Esquematiza el diseño de contenidos generales y carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de los contenidos. En otras palabras, se enfoca en "qué hace el producto, no cómo se ve."



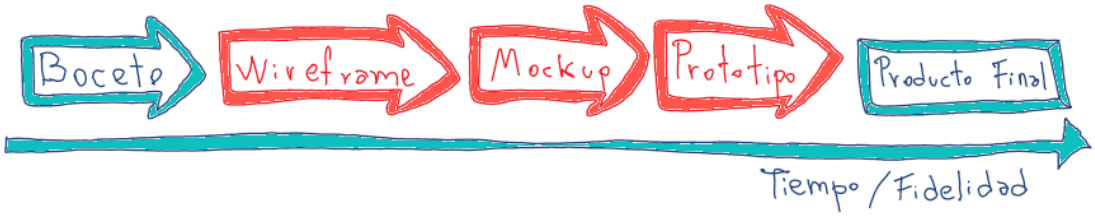
Mockup o maqueta

Proporciona detalles visuales, como colores y tipografía para dar al cliente una impresión más realista de cómo se verá el producto final. Se parece más a un producto terminado o prototipo, pero no es interactivo y no se puede hacer clic en él.



Prototipo

Está muy cerca del producto terminado donde los procesos pueden simularse y la interacción del usuario puede probarse. Se parece mucho al producto terminado y es una herramienta excelente para obtener comentarios de los usuarios y probar el producto antes de su lanzamiento.



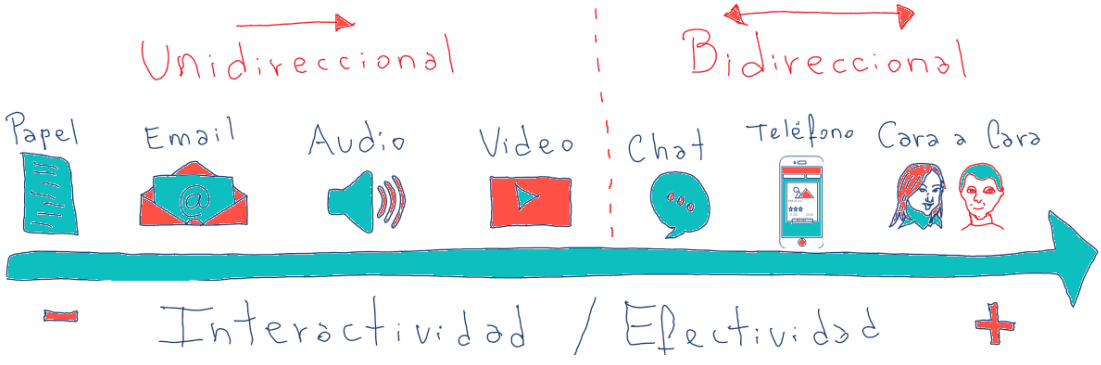
3.3 Comunicaciones

La clave de todo proyecto exitoso es la comunicación.

La interactividad y eficacia de los distintos canales de comunicación hay que tenerlos en cuenta al momento de comunicarnos con los interesados.

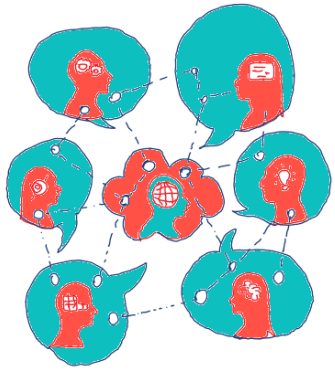
En los proyectos ágiles se recomienda siempre la comunicación cara a cara para promover la confianza y la comunicación bidireccional entre los miembros del equipo. Si el equipo trabaja de manera remota, estas comunicaciones deberían ser mediante videoconferencias u otros canales digitales que simulen la presencialidad cara a cara.

En el gráfico a continuación se indican algunos canales de comunicación desde los menos interactivos, menos efectivos y unidireccionales como el papel, hasta los más efectivos, más interactivos y bidireccionales como cara a cara.



Conocimiento compartido

Otro de los conceptos básicos de la filosofía ágil es la de compartir el conocimiento entre los miembros del equipo e interesados clave. De esta forma, si algún miembro del equipo no trabajara más en el proyecto, el resto del equipo puede seguir funcionando con normalidad hasta conseguir un reemplazo.



El conocimiento debe ser compartido entre el equipo, el cliente, la comunidad y la organización.

Compartir el conocimiento no implica solamente compartir información del proyecto como planes, avances o problemas; sino compartir las lecciones aprendidas sobre las mejores prácticas para realizar las actividades del proyecto.

El canal más utilizado para acelerar el conocimiento compartido es cuando el equipo está colocado en un ambiente "cara a cara" donde se puede aprender de la **comunicación osmótica**.

Cuando los miembros del equipo están trabajando en una misma sala, pueden captar la información relevante como si fuera por ósmosis. En otras palabras, pueden obtener información útil escuchando de "casualidad" lo que otros colegas están discutiendo, e involucrarse y colaborar cuando sea necesario.

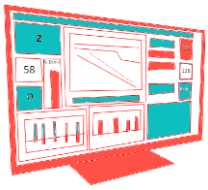
Los equipos distribuidos en diferentes lugares físicos podrían simular la comunicación osmótica con tecnología. Por ejemplo, cuando dos miembros del equipo tienen una reunión virtual, el resto del equipo podría estar escuchando esa conversación desde sus computadoras, aunque sigan trabajando en otras tareas. Cuando escuchen algún tema relevante en el que pueden aportar información, se unen a esa conversación.

Se suelen planificar reuniones para compartir el conocimiento entre el equipo y los interesados, como por ejemplo la reunión de revisión de la iteración.

Una de las herramientas para compartir el conocimiento entre los interesados es mediante la utilización de radiadores de información.

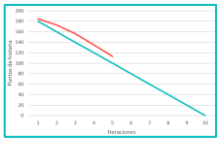
Radiador de información

Los radiadores de información son tableros escritos a mano, impresos o electrónicos que un equipo coloca en una ubicación altamente visible, para que todos los miembros del equipo, así como los interesados clave, puedan ver la información más reciente del proyecto de un vistazo. Por ejemplo:

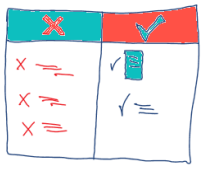


Un tablero electrónico de acceso remoto visible en el sitio principal de la organización donde se presenta en tiempo real el número de pruebas automatizadas, la velocidad del equipo, los informes de incidentes, el estado de integración continua, el listado de trabajo pendiente, tendencias, riesgos, etc.

Un tablero Kanban con tarjetas y gráficos dibujados a mano que utiliza baja tecnología y alto contacto (low tech – high touch).



Un diagrama de trabajo pendiente (Burndown chart) indicando las tareas que faltan completar a través del tiempo.



Registros de problemas visualizando los impedimentos que se han resuelto y los que están pendientes de resolución.

Los radiadores de información son de utilidad cuando:

- Los interesados se detienen para leer la información del proyecto.
- Los miembros del equipo no se quejan por tener que actualizar la información.
- Reflejan la realidad del proyecto en tiempo y forma.
- El equipo no tiene nada que ocultar a los interesados y se reconocen los problemas.
- Son simples, inequívocos, actuales, influyentes y altamente visibles.

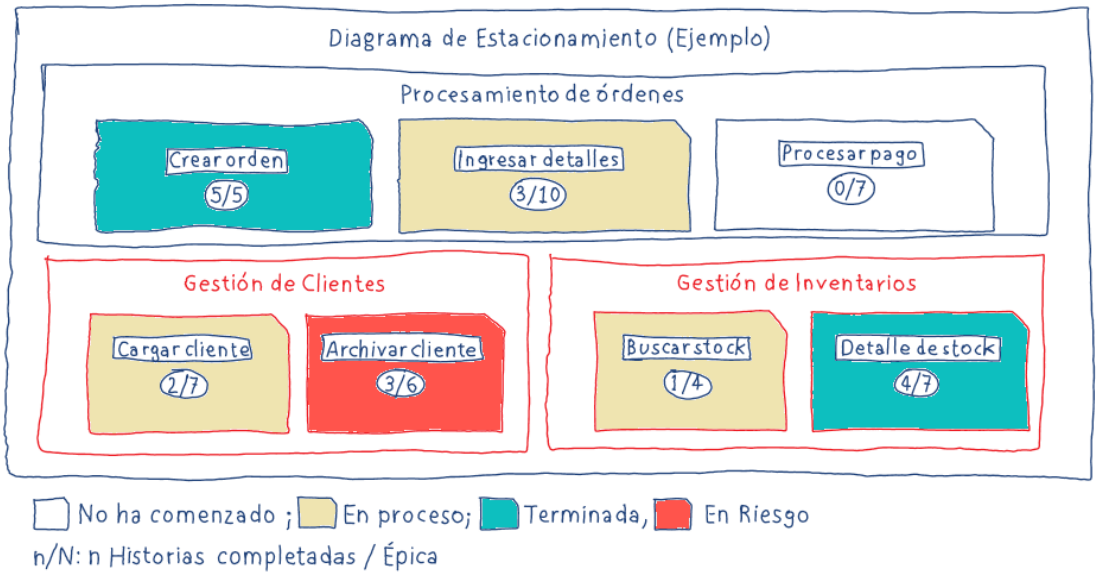
Los radiadores de información se utilizan para gestionar de manera proactiva las expectativas de los interesados y proporcionar transparencia sobre el trabajo que se realiza. Suelen provocar conversaciones con interesados externos al proyecto, lo que puede generar ideas útiles para la mejora continua.

Diagrama de estacionamiento

Un diagrama de estacionamiento (Parking lot) presenta el estado de avance de los bloques de funcionalidad de un producto. Se suele utilizar para comunicar el estado de avance de productos complejos en los que se trabaja en varias áreas y épicas al mismo tiempo.

Proporciona el estado del proyecto de un vistazo, lo que permite a los interesados en el producto realizar preguntas para obtener más información y detectar rápidamente si se está trabajando en la dirección correcta.

El gráfico utiliza cuadros para indicar la relación entre los diferentes niveles en la jerarquía de funcionalidad del producto y colores para indicar el estado de cada épica. Se llama diagrama de estacionamiento porque los cuadros se asemejan a una representación gráfica de un estacionamiento para autos.

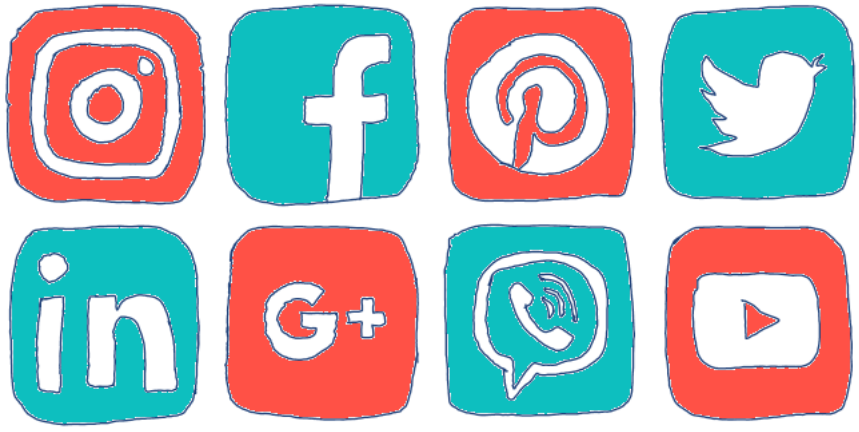


Cuando una épica cumple el objetivo por el cuál se decidió su desarrollo, aunque no hubieran finalizado todas las historias de usuario planificadas inicialmente para esa épica, podría considerase como “terminada”.

Redes sociales

La gestión exitosa del proyecto requiere formar y mantener relaciones entre los interesados a través de comunicaciones de calidad. La comunicación de calidad se logra cuando los interesados del proyecto pasan suficiente tiempo comunicándose entre sí, intercambiando información tanto formal como informalmente.

Las redes sociales suelen facilitar las comunicaciones de buena calidad y reducen el tiempo requerido para las reuniones cuando los miembros del proyecto son expertos de diversos campos y están localizados en diferentes países.



El uso de las redes sociales en proyectos ágiles no reemplaza a las reuniones cara a cara, pero complementa las comunicaciones cuando hay un gran número de personas involucradas.

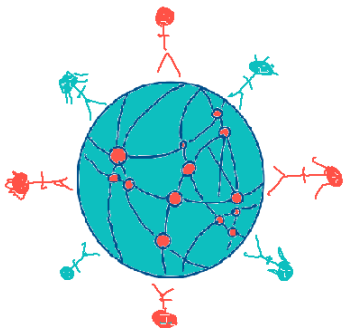
Antes de fomentar el uso de redes sociales para gestionar el proyecto, es necesario establecer reglas de seguridad para mitigar el riesgo de interceptación de datos, fraudes de información y espionaje.

Algunos de los beneficios de las comunicaciones basadas en redes sociales son los siguientes:

- Facilita la distribución de información cuando los interesados están distribuidos globalmente y son de diferentes generaciones
- Permite socializar ideas o prototipos a un bajo costo invitando a los usuarios a enviar su retroalimentación (ideas, requisitos, comentarios, etc.), antes de avanzar con el desarrollo del producto final

Diversidad de equipos

Cada vez hay más proyectos ágiles que se gestionan con equipos distribuidos globalmente. Esto podría complicar las comunicaciones entre los interesados al tener que gestionar diferentes culturas, lenguajes, edades, terminologías, etc.



Por ejemplo, mientras que en la cultura del continente Americano o Europeo es normal levantar la mano durante una reunión para comentar críticas negativas para mejorar el proyecto, en algunas culturas Orientales esto sería muy poco probable que ocurra porque sería faltarle el respeto al líder del proyecto.

Los equipos multiculturales proporcionan diversos conjuntos de habilidades. Estas diferentes experiencias podrían aprovecharse mediante una comunicación efectiva para mejorar el rendimiento del proyecto.

Algunas acciones para gestionar las comunicaciones con equipos multiculturales:

Realice una reunión al inicio del proyecto cara a cara, para que el equipo pueda conocerse mejor. Lo más importante de esa reunión será lograr algún tipo de socialización entre los miembros del equipo y los interesados clave.

Mantener reuniones privadas uno a uno, antes de la reunión formal con todo el equipo, para obtener un feedback más sincero de los diferentes interesados.

Realice una reunión de planificación de lanzamiento cara a cara involucrando la participación del equipo y los interesados clave para mejorar la colaboración.

Si el equipo tiene distintos usos horarios (ej. Brasil, Suiza, Australia e India), rotar el horario de las reuniones para distribuir el esfuerzo de trabajar en horarios anormales.

Capacitar sobre diferencias culturales, donde los equipos puedan comprender el comportamiento aceptable y no aceptable.

Capacitar sobre la construcción de equipos e interrelaciones personales para conocer la personalidad y motivaciones del equipo y los interesados.

Crear una cultura dónde informar problemas sea una buena práctica y nadie será culpado por ello.

Participación de interesados en las reuniones

Cada metodología ágil puede recomendar un enfoque diferente sobre cómo y cuándo deberían participar los interesados externos al equipo en las diferentes reuniones del proyecto, por lo que no es simple realizar un resumen que aplique para todos los casos.

Sin embargo, en la tabla a continuación hemos intentado resumir el tipo de participación de los interesados en diferentes reuniones de un proyecto ágil.

Reunión	Tipo de participación de los interesados externos al equipo	
Estimación de historias	Alta ★★★	Fundamental en las discusiones preliminares sobre los requisitos y la definición de las historias de usuario.
Priorización del product backlog	Media ★★	Mantener una comunicación fluida para que provean información antes que el dueño del producto decida las prioridades.
Planificación de la iteración	Media ★★	Podrían participar para aportar información al equipo sobre dependencias, riesgos, oportunidades, recursos compartidos, valor, etc.
Diaria	Baja ★	Podrían asistir a la reunión, pero sin participar. Cualquier consulta o aporte, la debería realizar al líder ágil después de la reunión diaria.
Revisión de la iteración	Alta ★★★	Fundamental para escuchar su retroalimentación sobre el producto y mantenerlos involucrados para mejorar la calidad del proyecto.
Retrospectiva	Baja ★	Podrían participar algunos interesados clave vinculados al proyecto.

3.4 Colaboración

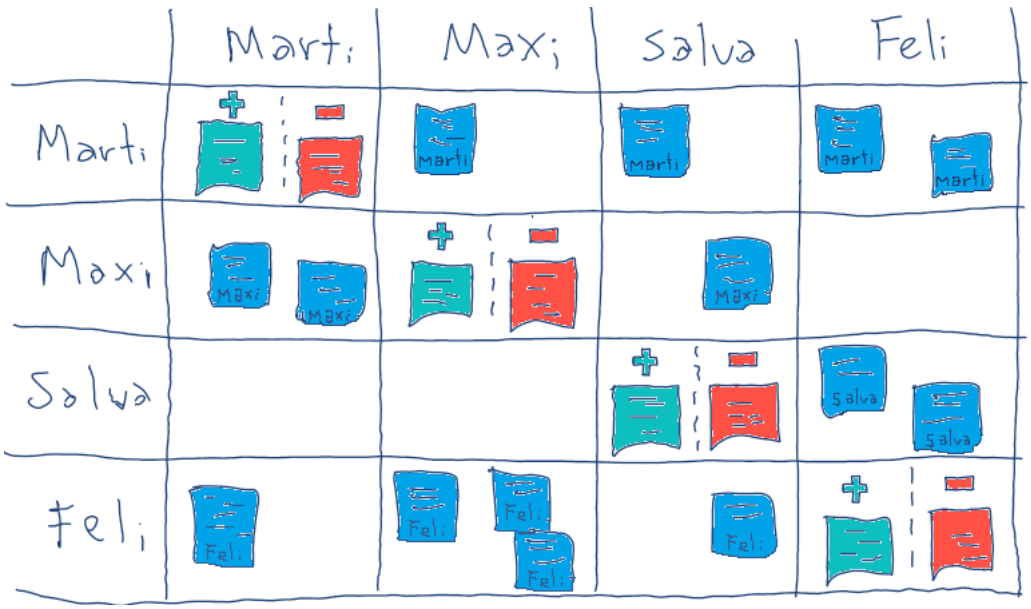
La colaboración es trabajar con otras personas en un objetivo común para alcanzar mejores resultados de lo que cualquier persona podría lograr por sí solo.

Para que la colaboración sea efectiva, los miembros del equipo e interesados necesitan tener buenas habilidades de coordinación y comunicación.

Matriz de colaboración

La matriz de colaboración es una tabla de doble entrada donde se colocan los nombres de todos los miembros del equipo en la primera fila y se repiten los nombres en la primera columna. El facilitador o coordinador del proyecto no forma parte de esta tabla.

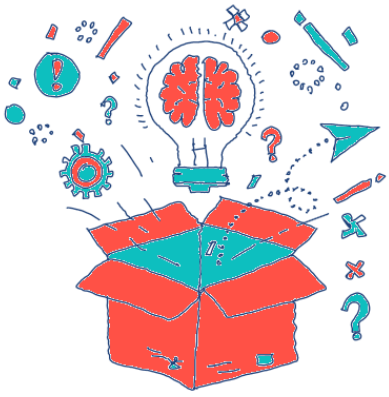
La diagonal de la tabla, donde coincide el nombre del miembro del equipo, la subdividimos en dos partes, la primera la marcaremos con el signo **+** y la segunda con el signo **-**.



En primer lugar, le pediremos a los miembros del equipo que en un post-it verde escriba lo que quiere hacer más y en un post-it rojo escriba lo que quiere hacer menos. Luego, en post-it azules cada miembro del equipo rellena las acciones que puede hacer para ayudar a su compañero a hacer más de lo que quiere hacer, o menos de lo que no quiere hacer. Finalmente, tenemos todo lo que el equipo puede hacer para ayudar a sus colegas.

Tormenta de ideas

La tormenta de ideas (o lluvia de ideas) es una herramienta de trabajo colaborativo que facilita el surgimiento de nuevas ideas para identificar alternativas, resolver problemas o mejorar procesos. Por ejemplo:



- Definir las funcionalidades que debería incluir el producto viable mínimo
- Estimar el esfuerzo necesario de las historias de usuario
- Identificar las oportunidades y amenazas que enfrenta el proyecto
- Discutir diferentes soluciones técnicas a un problema

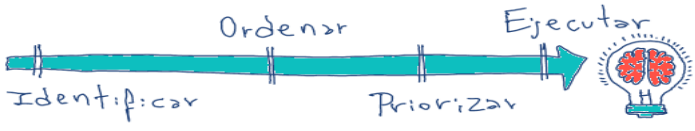
Algunas reglas básicas para llevar a cabo la dinámica de la tormenta de ideas:

REGLAS

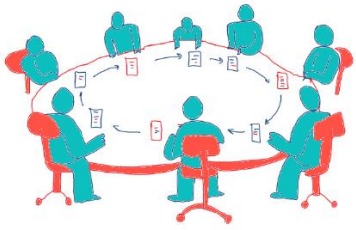
- Todas las ideas son válidas y nadie puede criticarlas al inicio de la sesión. Reservar las críticas para más adelante cuando se haya ampliado la discusión y las opciones.
- Pensar libremente sin ningún tipo de restricción para proponer hasta la idea más descabellada o impracticable. Las ideas innovadoras varias veces nacen de lo que parecía inicialmente una locura.
- Generar un gran número de ideas antes de comenzar a discutirlas.
- Sugerir mejoras a las ideas de otros o combinar distintas ideas en una nueva idea.

Proceso:

1. Identificar la mayor cantidad de ideas que sea posible
2. Agrupar las ideas similares y eliminar las que estén repetidas
3. Priorizar las ideas
4. Implementar las mejores ideas



Round-Robin: técnica de colaboración para fomentar la lluvia de ideas.



Paso 1: los miembros del equipo sentados alrededor de una mesa escriben individualmente sus ideas en tarjetas sobre algún problema que quieran resolver.

Paso 2: cada persona entrega la tarjeta con su idea a la persona que está al lado.

Paso 3: cada persona usa la idea de su vecino como inspiración para crear otra idea sobre la misma tarjeta y vuelve a pasar la tarjeta en ronda.

Paso 4: continúa el intercambio circular de ideas durante el tiempo establecido para luego reunir y discutir todas las ideas.

Scamper: lluvia de ideas en equipo que se utiliza para desarrollar o mejorar productos. Se basa en que para generar algo original no tiene por qué ser algo nuevo, sino que se puede desarrollar creatividad a partir de cambios o combinaciones de ideas ya existentes.

Sustituir: ¿Se pueden reemplazar componentes, materiales, personas?

Combinar: ¿Se pueden combinar otras funciones, dispositivos, etc.?

Adaptar: ¿Se pueden adaptar nuevos elementos o funciones?

Modificar: ¿Se puede modificar el tamaño, la forma, el color u otro atributo?

Propósito: ¿Podría darle al producto un uso diferente o usarlo en otra industria?

Eliminar o minimizar: ¿Hay funcionalidades que se pueden eliminar o simplificar?

Reorganizar: ¿Qué se puede intercambiar?

Crazy 8: Los miembros del equipo deben dibujar ocho ideas distintas en pocos minutos para generar una amplia variedad de soluciones a un problema.

Escritura de ideas (Brainwritting): dinámica de grupo similar al brainstorming donde los participantes escriben en un papel sus ideas antes de compartirlas con los demás. Esta técnica busca aumentar la participación de todos los interesados, en especial de los más tímidos que podrían no participar en un brainstorming.

Juegos de colaboración [12]

Los juegos de innovación (Innovation Games® - Luke Hohmann) son juegos colaborativos que involucran a clientes y partes interesadas para revelar lo que realmente les importa de un producto o servicio.

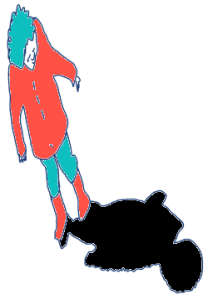
Estos juegos se utilizan para trabajar de manera colaborativa con el equipo y los clientes para: obtener retroalimentación sobre el producto a desarrollar, definir las funcionalidades, priorizar las historias de usuario, evaluar alternativas, etc.

1. **Caja de productos**: los clientes trabajan individualmente o en pequeños equipos para crear y vender su producto ideal.

Objetivo: identificar las características más interesantes del producto.

Actividad: los clientes deben imaginar que están vendiendo su producto en una feria comercial. En una caja de cartón deben diseñar las características del producto: marca, logotipo, slogan, características principales, detalles, requisitos, beneficios, condiciones de uso, etc. Cuando terminen la caja, finja ser un comprador potencial escéptico y pídales a los clientes que utilicen esa caja para venderle el producto.

2. **Mi sombra y Yo**: descubra las necesidades ocultas observando cuidadosamente lo que los clientes realmente hacen con sus productos.



Objetivo: identificar las necesidades ocultas del cliente.

Actividad: los observadores registran cuidadosamente lo que hace un cliente mientras utiliza el producto. Los observadores se sientan al lado del cliente para observar y escuchar acciones, expresiones, comentarios y sugerencias. Los observadores hacen preguntas al participante, como "¿Por qué estás haciendo eso?" o "¿Qué estás pensando en este momento?".

[12] Luke Hohmann, Innovation Games: Creating breakthrough products through collaborative play.

3. Compre una característica: los clientes trabajan juntos para comprar las características del producto más deseadas.

Objetivo: priorizar características.

Actividad: se crea un listado de características del producto con un costo estimado para cada una. A los participantes se les otorga dinero para comprar, pero ninguno tendrá dinero suficiente para comprar por sí solo una característica. Los participantes deberán negociar entre ellos para priorizar qué características serán las que puedan comprar.

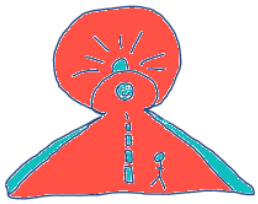
4. Bríndeles un jacuzzi: los clientes brindan comentarios sobre las características escandalosas para establecer lo que es realmente esencial.

Objetivo: utilice características escandalosas para descubrir avances ocultos.

Actividad: cada una de las características potenciales del producto aparecen en tarjetas barajadas. Algunas de las características propuestas son completamente escandalosas, como un celular que se utilice para freír huevos, una notebook que sirve para planchar el pelo, o una oficina que tenga un jacuzzi en la sala de recepción. Los observadores notan lo que sucede cuando un cliente descubre una de estas características escandalosas.

Resultados: Una característica verdaderamente escandalosa induce disonancia cognitiva que incomoda a los clientes. Entonces, trabajan mentalmente para deshacerse de la incomodidad rechazando la característica porque nunca se discutió realmente. Sin embargo, hay veces que transforman esa característica "escandalosa" en algo que no es tan escandaloso y este tipo de transformaciones son los que crean avances innovadores en el producto.

5. Recuerde el futuro: comprenda la definición de éxito de sus clientes al ver cómo moldean su futuro.



Objetivo: comprender la definición de éxito de sus clientes.

Actividad: Sus clientes deben imaginar que están en el futuro y que han estado usando su producto continuamente entre ahora y esa fecha futura (ejemplo: 6 meses). Pídale a su cliente que escriba exactamente lo que su producto habrá hecho para que sea feliz (exitoso, rico, seguro, inteligente, generoso, etc.).

6. Telaraña: los participantes trabajan individualmente o en pequeños equipos para crear imágenes vívidas de cómo sus productos y servicios encajan en su mundo.

Objetivo: comprender las relaciones de nuestro producto con otros productos.

Actividad: ponga el nombre de su producto en el centro de un círculo. Solicite a sus clientes que dibujen otros productos o servicios relacionados con su producto y pídales que le digan cuándo, cómo y por qué se usan. Vincular con líneas los diferentes productos y servicios.

7. Comience su día: los participantes describen en colaboración cuándo, cómo y dónde usan sus productos durante una jornada (día, semana, mes, año)

Objetivo: comprender cuándo y cómo su cliente utiliza su producto.

Actividad: pídale a su cliente que describa los eventos diarios, semanales, mensuales y anuales relacionados con el uso de su producto en un calendario o línea de tiempo dibujada en un papel. Identifique cómo su producto ayuda o dificulta el día de su cliente.

8. **Mostrar y decir**: los clientes describen los artefactos producidos por su sistema.

Objetivo: identificar los artefactos creados por su producto.

Actividad: Solicite a sus clientes que le muestren ejemplos de artefactos creados gracias a su producto y que le digan por qué estos artefactos son importantes, incluyendo cuándo y cómo se usan. Por ejemplo, si su producto es un software para imprimir 3D, pídales que le muestren algún prototipo impreso.

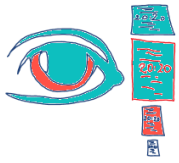
Resultados: descubrir que ciertos artefactos esperados por el producto no se están realizando, o que existen artefactos sin utilidad, o artefactos que se usan de manera inesperada.

9. **El aprendiz**: el desarrollador trabaja en la industria del producto que quieren desarrollar.

Objetivo: crear empatía con el cliente.

Actividad: Solicite a los desarrolladores de productos que se pongan en los zapatos del cliente conociendo más sobre el producto que están desarrollando. Por ejemplo, si están desarrollando un software para otorgar préstamos, pídales que trabajen un tiempo otorgando préstamos; o si están desarrollando una app para mejorar ventas online, pídales que trabajen un tiempo vendiendo productos online. De esta forma los desarrolladores obtienen un mejor conocimiento de la experiencia del cliente y cierto grado de empatía por el problema que intentan resolver.

10. **Visión 20/20**: los clientes negocian la importancia relativa de aspectos como las características del producto, los requisitos del mercado y los beneficios del producto.



Objetivo: comprender las prioridades del cliente.

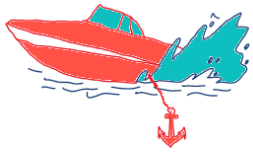
Actividad: escriba las características del producto en tarjetas. Baraja el montón y ponlos boca abajo. Tome el primero de la parte superior de la pila y póngalo en la pared. Tome el siguiente y pregunte a sus clientes si es más o menos importante que el que está en la pared. Si es más importante, colóquelo más arriba; si es menos importante, póngalo más abajo. No se permite que dos características sean de igual importancia. Repita este proceso con todas sus tarjetas de características y desarrollará una visión 20/20 de las prioridades del cliente.

11. Pode el árbol de productos: los clientes trabajan en pequeños equipos para dar forma a la evolución de sus productos y servicios.

Objetivo: moldear su producto a las necesidades del mercado.

Actividad: dibuje un árbol en una pizarra. Las extremidades gruesas representan las principales áreas de funcionalidad dentro de su sistema. Las ramas externas representan las características disponibles en la versión actual. Escriba nuevas características potenciales en tarjetas en forma de hojas. Solicite a sus clientes que coloquen las características deseadas alrededor del árbol, definiendo la siguiente fase de su crecimiento. ¿El árbol crece de manera equilibrada? ¿Alguna característica obtiene la mayor parte del crecimiento? ¿Se vuelve más fuerte un aspecto subutilizado?

12. Lancha rápida: los clientes identifican sus mayores puntos débiles con sus productos



Objetivo: identificar lo que a los clientes no les gusta de su producto.

Actividad: dibuje una lancha en una pizarra. Los clientes escriben lo que no les gusta en anclas sobre la lancha que no le permitirán moverse rápido. Podrían estimar qué tan rápido iría la lancha cuando se eliminen anclas. Cuando los clientes terminen de publicar sus anclas, revise cada una, confirmando cuidadosamente su comprensión de lo que quieren ver cambiado en el sistema.

13. Mi peor pesadilla: los participantes imaginan y dibujan una caricatura de su "peor pesadilla" relacionada con el producto que está investigando.

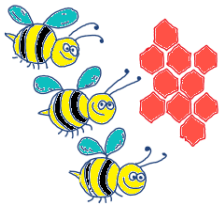
Objetivo: descubrir los peores escenarios ocultos o no considerados.

Actividad: Pídales a los participantes que dibujen su "peor pesadilla" relacionada con el producto que están investigando. Por ejemplo, si está desarrollando un software para transacciones bancarias, su “peor pesadilla” podría ser perder a todos sus clientes debido a un hacker. Después de completar las ilustraciones, pida a los participantes que presenten su "peor pesadilla" al grupo.

Responsabilidad compartida

Tanto el **enjambre** (swarming), el **acoso** (mobbing) y el **trabajo en pareja** (pairing) son técnicas de desarrollo de software donde el equipo trabaja de manera simultánea en una misma tarea con responsabilidad compartida. Hasta que los miembros del equipo no finalicen la historia de usuario que están desarrollando, no podrán avanzan a la próxima tarea. Todo el equipo tiene responsabilidad compartida para finalizar la historia de usuario lo antes posible.

Enjambre (Swarming)



Los miembros del equipo trabajan cada uno con su pc y teclado realizando las actividades en paralelo. Por ejemplo, trabajar el análisis, desarrollo y las pruebas en paralelo. Cuando el analista tiene un pequeño avance le comunica al desarrollador para que comience el desarrollo; cuando hay un pequeño avance con el desarrollo, el probador comienza con los casos de prueba.

Acoso (Mobbing)

El equipo se sienta alrededor de una mesa con un solo teclado e intercambian el teclado cada 15 minutos por lo que va cambiando el liderazgo. La salida del monitor podría ir a una pantalla grande para que todos puedan ver lo que está haciendo la persona que escribe.

Trabajo en pareja (Pairing)

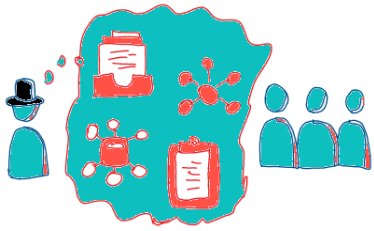
Dos miembros del equipo trabajan en la tarea compartiendo un solo teclado y cambiando roles entre el “conductor” que escribe y el “navegador” que observa y revisa cada línea de código a medida que se escribe. Los dos programadores cambian los roles con frecuencia. El conductor centra toda su atención en los aspectos "tácticos" de completar la tarea actual, mientras que el observador considera la dirección "estratégica" del trabajo y presenta ideas para mejorar.

Propiedad del código compartido: todos los miembros del equipo pueden realizar cambios en cualquier archivo de código según sea necesario. Por ejemplo, para completar una tarea de desarrollo, para reparar un defecto, o para mejorar la estructura general del código. Ventajas: reduce el riesgo de que la ausencia de un desarrollador ralentice el trabajo, aumenta la probabilidad de que el diseño general sea el resultado de decisiones técnicas acertadas y anima a cada desarrollador a sentirse responsable de la calidad del conjunto. Problema: cuando todos son responsables de la calidad del código, podría ocurrir que nadie sea responsable de la baja calidad en casos de mala praxis.

Especificación mediante ejemplos (SBE)

La especificación mediante ejemplos (SBE: Specification by Example) es un enfoque colaborativo para definir los requisitos de software basado en ejemplos, en lugar de enunciados abstractos.

El dueño del producto y los interesados del negocio proporcionan al equipo ejemplos reales sobre para qué se utilizará el software. Por su parte, el equipo de desarrollo, en colaboración con los interesados, debe asegurarse de que cada especificación sea comprobable.

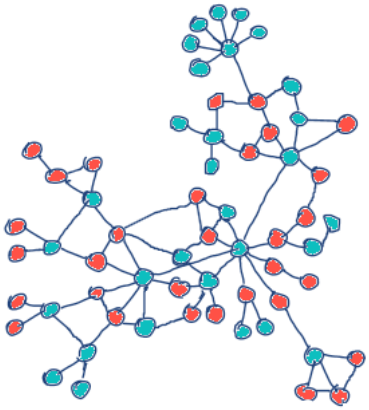


Los ejemplos no deberían centrarse en detalles técnicos de cómo funciona el software, sino en los objetivos comerciales que se quieren alcanzar.

Ventajas de SBE:

- Fomenta la comunicación entre el dueño del producto y los desarrolladores
- Alinea las especificaciones con las pruebas de aceptación del usuario
- Simplifica el diseño y reduce el código innecesario en el desarrollo
- Acelera la implementación al acortar los bucles de retroalimentación

Liderazgo colaborativo



Los proyectos ágiles son un sistema complejo adaptativo (CAS) de muchos individuos actuando en paralelo y aprendiendo de lo que los otros individuos realizan.

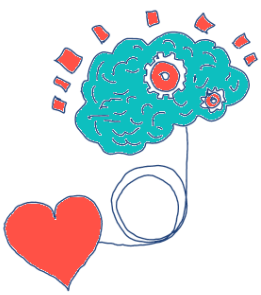
El control de un CAS tiende a estar muy disperso y descentralizado, lo que implica que un único líder del proyecto no puede coordinar las acciones de todos los demás individuos.

El comportamiento coherente del proyecto surge de la colaboración entre los miembros del equipo e interesados, con un liderazgo poliárquico que involucra a todos.

3.5 Habilidades blandas

Además de la comunicación y colaboración, para un involucramiento exitoso de los interesados son necesarias habilidades blandas tales como la inteligencia emocional, escucha activa, negociación, resolución de conflictos, etc.

Inteligencia emocional [13]



La inteligencia emocional es la capacidad de sentir, entender, controlar y modificar los estados emocionales de uno mismo y de los demás, promoviendo un crecimiento emocional e intelectual.

La inteligencia emocional se empieza a desarrollar durante la infancia y se puede seguir desarrollando a lo largo de toda la vida. El primer paso para desarrollar la inteligencia emocional es aprender a reconocer nuestras emociones y sentimientos para luego comprender de dónde provienen y poder controlarlas.

Componentes:

1. Autocontrol: autorregulación, pensar antes de hablar o de actuar, reflexionar y controlar nuestros impulsos;
2. Conocimiento de uno mismo: ser conscientes de nuestras emociones, saber qué nos duele y el por qué de ese sufrimiento, molestia o contradicción;
3. Automotivación: motivación intrínseca personal que genera energía positiva capaz de darnos aliento aún cuando lo que nos rodea no es satisfactorio;
4. Empatía: comprender los sentimientos y emociones de la otra persona, experimentando de forma objetiva lo que siente el otro;
5. Habilidades sociales: ser asertivos, buenos comunicadores, pacientes, abiertos, compasivos, positivos y saber respetarnos a nosotros mismos.

Por otro lado, la **inteligencia social** implica conocer el estado de los demás integrantes de la comunidad, entendiendo cómo el entorno puede afectar el desarrollo de las personas.

[13] Daniel Goleman, Inteligencia emocional.

Escucha activa [14]

Escuchar activamente a los demás es una de las habilidades más importantes de un buen líder o entrenador (coach). Lo primero que debe hacer para poder escuchar activamente es enfocarse en los gestos, lenguaje corporal, palabras y estado emocional de la otra persona para entender empáticamente la situación del otro.

Suelen haber 3 niveles de escucha:

Nivel 1 - Escucha interna: tiende a escuchar más a su propia voz interior, formando opiniones, en lugar de centrarse en lo que se dice. Estamos enfocados únicamente en nosotros mismos para reunir información y satisfacer nuestras propias necesidades. Sirve para recopilar datos, pero no para tener conversaciones profundas.

Nivel 2 - Escucha enfocada: mantiene un enfoque en lo que dice la persona, escuchando atentamente cada palabra y "escuchando" cada matiz de la conversación. La atención está en la otra persona para entender sus emociones. Se profundiza lo que sabe sobre la otra persona al mantener la curiosidad y hacer preguntas aclaratorias durante el diálogo.

Nivel 3 - Escucha global: descubrir el significado subyacente de las palabras ligadas a las emociones de la persona, lo que permite conectarse con la otra persona. Escuchar lo que ves, oyes, hueles y sientes de la otra persona. Escuchar todo el entorno de la persona, su lenguaje corporal, su energía y emociones.

Un coaching efectivo comienza en el Nivel 2 y tiene la capacidad de escuchar en el Nivel 3.

Consejos para escuchar activamente:

- Desházase de distracciones como celulares, computadoras, lugares ruidosos, etc.
- Mantenga una buena postura desplegando los brazos y un contacto visual
- Preste atención al orador y reconozca el mensaje verbal y no verbal
- Transmita atención, como asentir ocasionalmente y hacer preguntas
- Evite juicios o creencias que puedan distorsionar lo que se dice
- Diferir el juicio y no interrumpir con contraargumentos, opiniones y análisis
- Escuche hasta el final, sea sincero, honesto y respetuoso al responder

[14] Karen Kimsey-House y otros, Co-active coaching.

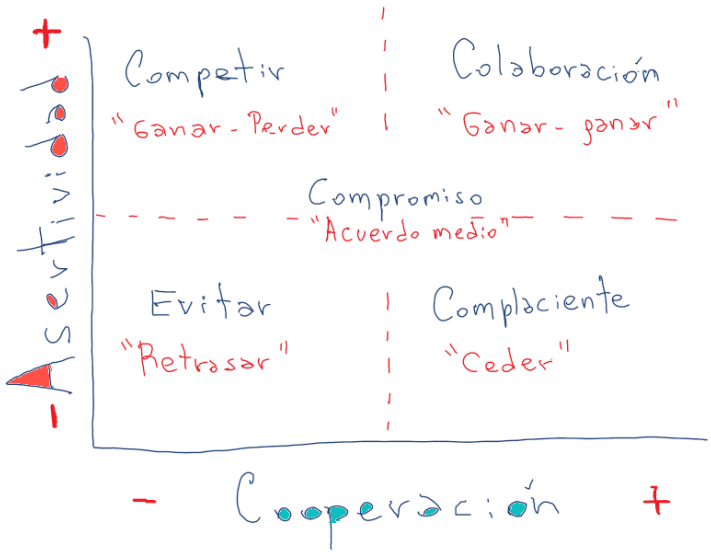
Negociación y resolución de conflictos

La negociación ágil se lleva a cabo cuando las partes en conflicto confían en un proceso para descubrir una solución mutuamente aceptable, que se considere justa y respetable para todas las partes. Existen varias técnicas sobre cómo llevar a cabo una negociación exitosa.

Consejos para tener en cuenta durante la negociación:

- ✓ Separar a las personas del problema: concentrarse en el problema que se quiere resolver y no en las personas. Los prejuicios sobre las personas involucradas en la negociación bloquean la consideración de alternativas.
- ✓ Concéntrese en los intereses y no en las posiciones: el interés que está tratando de proteger o mejorar durante el proceso de negociación debe ser el objetivo de la negociación, no las posiciones de las personas involucradas.
- ✓ Generar opciones para el beneficio mutuo: tratar constantemente de pensar en alternativas donde haya un beneficio para ambas partes, para terminar con un acuerdo de colaboración "ganar-ganar".
- ✓ Utilice criterios objetivos: será más fácil lograr acuerdos sobre criterios objetivos en lugar de subjetivos.

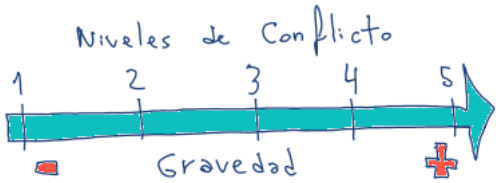
Estilos de conflicto [15]:



[15] Thomas–Kilmann conflict mode instrument (TKI).

Niveles de conflicto [16]

Los conflictos son inevitables e incluso deseables si los sabemos gestionar. El conflicto indica que los miembros del equipo se sienten libres de expresar diferentes puntos de vista, lo cual es una señal de un equipo saludable y causal de innovación. Sin embargo, si el conflicto no se gestiona correctamente, puede ser perjudicial para el equipo.

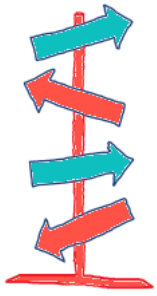


Cuanto mayor es el nivel de conflicto, más difícil será llegar a un acuerdo cordial.

	Conflicto	Resolución
Nivel 1	El equipo reconoce que hay un problema por resolver, discuten sobre la base de hechos y están colaborando y compartiendo información para resolver el problema.	Colaboración y consenso para una solución ganar-ganar.
Nivel 2	Desacuerdo donde los miembros del equipo se distancian para tratar de proteger sus propios intereses. Dejan de tomar la iniciativa para resolver el problema por temor a que pueda comprometer su posición.	Llevar a las partes a una discusión donde por lo general los equipos llegarán a una solución.
Nivel 3	Contienda donde las personas comienzan a alinearse con un lado u otro con posiciones más rígidas donde quieren "ganar". Comienzan los ataques personales.	Ceder, negociar y obtener hechos.
Nivel 4	Cruzada donde los miembros del equipo creen que las personas del otro lado no cambiarán su posición, por lo que deben ser destruidas. Se hacen generalizaciones excesivas sobre el problema y sobre las personas. Las partes en conflicto no se sentarán a discutir en una misma mesa.	Diplomacia para hablar con cada parte por separado para tratar de reducir las diferencias y volver a un nivel más bajo de conflicto.
Nivel 5	Guerra donde el problema es irresoluble. No es suficiente que uno gane, los otros deben perder. No habrá resultado constructivo. Casi todas las vías de reconciliación están cerradas y se intercambia muy poco sin diálogo.	Limitar o contener el daño para evitar que las personas se lastimen unas a otras.

[16] Speed Leas, Levels of conflict.

3.6 Toma de decisiones



Los miembros del equipo y los interesados deben tomar decisiones eficientes que permitan la celeridad del proyecto. En los proyectos ágiles, cualquier retraso en la toma de decisiones suele tener un mayor impacto que en proyectos de ciclos predictivos.

Lo primero que deberíamos hacer es establecer el **marco de decisión** (decision framing) identificando a quiénes involucraremos en el proceso de decisión. Quien toma la decisión es menos importante que involucrar a las personas adecuadas en el proceso de decisión. Por ejemplo, el dueño del producto podría tomar malas decisiones sin el aporte de los desarrolladores y los interesados clave del proyecto

Las herramientas de **facilitación** permitirán que una discusión termine en una decisión. El proceso de facilitación se basa en reuniones en equipo para aclarar expectativas y trabajar de manera colaborativa. Por ejemplo, herramientas colaborativas como la caja de producto, telaraña, comprar una característica, visión 20/20, lancha rápida, etc., facilitarán el involucramiento de los interesados en la toma de decisiones.

Toma de decisiones participativa

La toma de decisiones participativa consiste en involucrar al equipo e interesados clave en la toma de decisiones. Esto no significa tomar todas las decisiones por consenso, sino involucrar a las personas apropiadas en el proceso de toma de decisiones para que todos los puntos de vista sean considerados antes de tomar una decisión.

Modelos de toma de decisiones participativas:

Basado en insumos (consulta): todos los participantes tienen la oportunidad de brindar aportes en el proceso de toma de decisiones.

Colaboración compartida (consenso): los participantes no sólo son consultados, sino que también se involucran activamente para llegar a una decisión consensuada. Una decisión por consenso no implica que todos están 100% de acuerdo, sino que los interesados aceptan la decisión tomada por el grupo.

Comando: la decisión final la toma un líder de alto rango o un pequeño grupo de personas y los miembros del equipo son informados sobre la decisión.

Empoderar al equipo

Los métodos ágiles favorecen más el empoderamiento del equipo y menos la dirección de comando y control. Esto aumenta la satisfacción del equipo y la productividad, pero plantea la necesidad de una toma de decisiones efectiva para evitar retrasos del proyecto.

El líder servicial empodera al equipo y considera a todos los miembros por igual para intentar resolver las necesidades de cada uno de ellos, abordando el trabajo del proyecto en este orden:

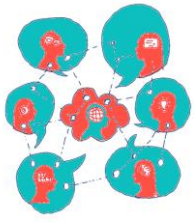
1º Propósito: definir con el equipo el "por qué" para que puedan comprometerse con el objetivo del proyecto.

2º Personas: motivar al equipo a crear un ambiente donde todos puedan tener éxito.

3º Proceso: no planificar un proceso ágil "perfecto", sino agregar valor frecuentemente y reflexionar sobre el producto y el proceso de manera colaborativa para la mejora continua.

Decisiones en equipo

Si no consultamos las opiniones de los miembros del equipo e interesados clave, corremos el riesgo de falta de compromiso y participación. Además, perderemos la oportunidad de considerar nuevas perspectivas innovadoras para el desarrollo de un mejor producto.



Darles a los interesados la oportunidad de registrar su preocupación es un componente importante para lograr un acuerdo para avanzar con el proyecto, respetando al mismo tiempo las opiniones discrepantes y manteniendo a todos comprometidos.

Algunas técnicas simples para fomentar y facilitar la participación de los interesados en el proceso de toma de decisiones suelen ser: votación simple; pulgares arriba-abajo-de lado; puño a cinco y el espectro de decisiones de Jim Highsmith.

Votación simple

La votación simple "a favor" (levantando la mano) o "en contra" (sin levantar la mano) es fácil y rápida. Este tipo de votación obtiene un resultado demasiado rápido y puede perder la oportunidad de una tercera alternativa con sugerencias de mejora para refinar la decisión.

Pulgares arriba, abajo o de lado

Mostrar los pulgares hacia arriba, hacia abajo o de lado es una forma rápida de lograr un voto simple. A las personas con el pulgar hacia los lados se les pregunta por qué están indecisos. Algunas veces son simplemente neutrales, otras veces tienen un conflicto, preocupación o preguntas que necesitan más investigación. Este enfoque es más rápido que sondear a todos para obtener información, ya que la mayoría de las personas no tendrán preocupaciones y solo querrán seguir adelante.

Puño a cinco

El mismo método que vimos para priorización, se puede utilizar para la toma de decisiones en equipo. Se vota con puño o entre 1 a 5 dedos para expresar el grado de acuerdo en la decisión.

Puño: desea bloquear la propuesta porque cree que es perjudicial.

Un dedo: tiene reservas serias, pero no bloqueará la iniciativa. Se compromete a abrir la comunicación con respecto a sus reservas.

Dos dedos: tiene reservas importantes, pero apoyará la iniciativa.

Tres dedos: apoya la iniciativa.

Cuatro dedos: brinda un fuerte apoyo y participación, pero no está dispuesto a liderar la iniciativa.

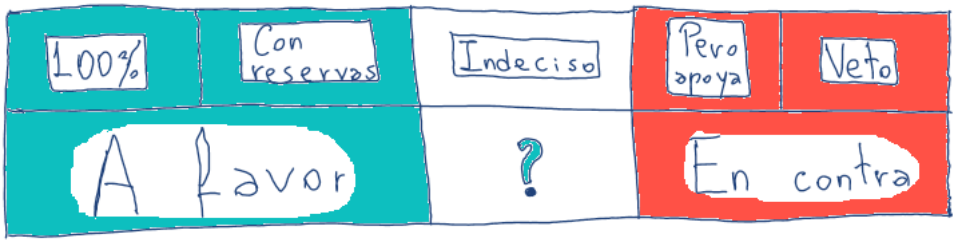
Cinco dedos: lo considera como la mejor idea, tomará la iniciativa si se lo piden.

¡Cuidado! Algunas organizaciones utilizan este mismo método con la escala inversa. O sea, puño es bueno porque no bloqueará la decisión y 5 dedos significa la máxima oposición a esa decisión.

Espectro de decisiones de Jim Highsmith [17]

Herramienta para tomar decisiones donde se les pide a los miembros del equipo que indiquen como se sienten en relación con una decisión:

- Totalmente a favor
- De acuerdo, pero con reservas
- Indeciso, con sentimientos opuestos
- En contra, pero apoyará la iniciativa
- Totalmente en contra y vetará la iniciativa



Las personas que no están totalmente a favor son invitadas a compartir sus preocupaciones. El solo hecho de poder expresar diferencias es suficiente para permitir que las personas se involucren en las decisiones del proyecto.

Esta herramienta permite a las personas indicar su apoyo a una decisión y expresar sus reservas al mismo tiempo.

[17] James Highsmith, Agile Project Management.

3.7 Ejercicios – Interesados

1 Marque quiénes serían interesados de un proyecto que utiliza metodologías Scrum.

Cliente	
Dueño del producto	
Equipo de desarrollo	
Patrocinador	
Scrum master	
Usuario	

2 Coloque del 1 al 5 los pasos para la gestión de interesados.

Orden	Proceso
	Comunicar periódicamente el estado de avance del proyecto.
	Conocer las necesidades de los interesados y los requisitos del proyecto.
	Gestionar activamente las expectativas de los interesados.
	Identifique periódicamente a los interesados.
	Involucrar a los interesados en el proyecto mediante la comunicación activa.

3 ¿Cuáles son los 3 principales componentes del diagrama de caso de uso?

4 Marque verdadero o falso

En relación con una "persona" en un proyecto ágil…	V o F
Ayudan a guiar las decisiones sobre la funcionalidad y diseño del producto.	
La persona principal podría quedar satisfecha con una funcionalidad diseñada para otra persona.	
Podemos inventar personas cuando no tenemos tiempo para una investigación.	
Representa a un individuo ficticio, que se basa en usuarios reales.	
Se describe en una ficha su nombre real, su imagen de rostro, sus gustos, etc.	

5 Marque los componentes mínimos de un acta de constitución de un proyecto ágil al inicio de un proyecto.

Componente	
Antecedentes	
Criterios de éxito	
Hitos	
Interesados	
Misión	
Objetivos	
Restricciones	
Supuestos	
Visión	

6 Marque los componentes mínimos que se incluyen en la caja visión del producto con formato de una hoja.

Componente	
Grupo objetivo	
Hitos	
Necesidad	
Objetivos	
Producto	
Restricciones	
Supuestos	
Visión	

7 Marque a qué herramientas corresponden estos contenidos: *Para (cliente objetivo), quien (necesita hacer qué) el (producto o servicio) es un (categoría de producto) que (beneficios clave). A diferencia de (competencia), nuestro producto (diferenciación).*

Herramienta	
Acta de constitución	
Caja de visión	
Discurso del ascensor	
Esquema de página	
Hoja de ruta del producto	
Product backlog	

8 Complete con las palabras correctas. Opciones: Actividades, Épicas, Historias de usuario, Misión, Objetivos, Tareas, Temas.

La hoja de ruta del producto se estructura en términos de _____ y _____ para reflejar la visión de largo plazo del proyecto.

9 Marque verdadero o falso

En relación con "el listado de trabajo pendiente asociado al producto" …	V o F
El dueño del producto debe involucrar al equipo y a los interesados claves para que den su opinión sobre el producto a desarrollar.	
Está en constante evolución y permanece disponible para que el equipo y los interesados contribuyan con sugerencias.	
El equipo de desarrollo y el dueño del producto realizan un refinamiento continuo agregando detalles y estimaciones sobre las características a desarrollar.	
Las historias de usuario que definan los interesados durante la iniciación se mantendrán estables a lo largo de todo el proyecto.	

10 Marque si son ejemplos de DoD (Definición de terminado) o DoR (Definición de listo)

Ejemplos	DoD o DoR
Criterios de aceptación y dependencias de la historia de usuario definidos	
El código cumple con los estándares de desarrollo	
Historia de usuario clara, comprobable y factible	
Los cambios de configuración fueron documentados y comunicados	
Persona que aceptará la historia de usuario identificada	
Pruebas unitarias escritas y aprobadas	
Todos los elementos del código tienen que estar completados	

11 Complete si las siguientes ítems corresponden a “wireframe”, “mockup” o “prototipo”.

Ítem	W, M o P
Esqueleto o estructura visual de un producto (ej. página web, app, o productos basados en pantallas).	
Esquematiza el diseño de contenidos generales y carece de estilo tipográfico, color o aplicaciones gráficas.	
Está muy cerca del producto terminado donde los procesos pueden simularse y la interacción del usuario puede probarse.	
Proporciona detalles visuales, como colores y tipografía para dar al cliente una impresión más realista de cómo se verá el producto final.	
Se enfoca en “qué hace el producto, no cómo se ve”.	
Se parece a un producto terminado, pero no es interactivo y no se puede hacer clic en él.	
Se parece mucho al producto terminado y es una herramienta excelente para obtener comentarios de los usuarios y probar el producto.	
Su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de los contenidos.	

12 Ordene los siguientes canales de comunicación desde los menos interactivos y efectivos (# 1) hasta los más interactivos y bidireccionales (# 7).

Canal de comunicación	Interactividad (1 baja a 7 alta)
Audio	
Cara a cara	
Carta	
Chat	
Email	
Teléfono	
Video	

13 ¿En cuál de los siguientes canales se da la comunicación osmótica?

Canal de comunicación	
Audio	
Cara a cara	
Carta	
Chat	
Email	
Teléfono	
Video	

14 Marque cuáles serían ejemplos de radiadores de información

Radiador de información	
Diagrama de trabajo pendiente	
Registros de problemas	
Tablero Kanban	
Teleconferencia	
Videoconferencia	

15 Complete verdadero o falso.

En relación con “el diagrama de estacionamiento” …	V o F
Comunica el estado de avance de productos complejos en los que se trabaja en varias áreas y épicas al mismo tiempo.	
Hasta que una épica no complete todas las historias de usuario planificadas inicialmente, no podrá considerase como “terminada”.	
Los cuadros se asemejan a un estacionamiento para autos.	
Permite a los interesados en el producto realizar preguntas para obtener más información y detectar rápidamente si se está trabajando en la dirección correcta.	
Presenta el estado de avance de los bloques de funcionalidad de un producto.	

16 Complete verdadero o falso.

En relación con “las redes sociales” …	V o F
En proyectos ágiles, puede reemplaza a las reuniones cara a cara	
Facilita la distribución de información con equipos distribuidos globalmente	
Facilitan las comunicaciones de buena calidad	
Hay que fomentar su uso sin necesidad de protocolos	
Permite socializar ideas a un bajo costo	
Pueden reducir el tiempo requerido para las reuniones	

17 Complete verdadero o falso.

En relación con “equipos multiculturales” …	V o F
Capacitar sobre la construcción de equipos e interrelaciones personales para conocer la personalidad de los interesados.	
Crear una cultura dónde informar problemas no sea bien visto, en su lugar priorizar noticias alentadoras que suban la moral del equipo.	
Mantener reuniones privadas uno a uno, antes de la reunión formal con todo el equipo.	
Planificar el horario de las reuniones que vote la mayoría, aunque la minoría tenga que trabajar en horarios anormales.	

18 Complete la participación (alta, media, baja) de los interesados externos al equipo.

Tipo de reunión	Participación de interesados externos
Diaria	
Estimación de historias de usuario	
Planificación de la iteración	
Priorización del product backlog	
Retrospectiva	
Revisión de la iteración	

19 Marque las reglas básicas de la tormenta de ideas.

Regla	
Comenzar con pocas ideas para facilitar la reunión	
Proponer ideas descabelladas o impracticables	
Sugerir mejoras a las ideas de otros	
Todas las ideas son válidas y nadie puede criticarlas al inicio de la sesión	

20 Complete con número 1 a 7 el objetivo que corresponde con cada uno de los siguientes juegos colaborativos.

#	Juego	#	Objetivo
1	Caja de productos		Crear empatía con el cliente
2	Mi sombra y Yo		Descubrir avances ocultos
3	Bríndeles un jacuzzi		Descubrir los peores escenarios ocultos o no considerados
4	Mostrar y decir		Identificar las características más interesantes del producto
5	El aprendiz		Identificar las necesidades ocultas del cliente
6	Lancha rápida		Identificar lo que a los clientes no les gusta de su producto
7	Mi peor pesadilla		Identificar los artefactos creados por su producto

21 Complete cada ítem si corresponde a Swarming, Mobbing o Pairing

Ítem	S, M, P
Los miembros del equipo trabajan cada uno con su pc y teclado	
El equipo se sienta alrededor de una mesa con un solo teclado	
Cuando el analista, desarrollador y el probador trabajan en paralelo	
La salida del monitor va una pantalla grande para que todos puedan ver	
Dos miembros del equipo trabajan en la tarea compartiendo un solo teclado	
Dos programadores cambian los roles de conductor y navegador	
El conductor es "tácticos", mientras que el observador es "estratégico"	

22 Complete verdadero o falso.

En relación con la “especificación mediante ejemplos (SBE)” …	V o F
Acelera la implementación al acortar los bucles de retroalimentación	
Alinea las especificaciones con las pruebas de aceptación del usuario	
Fomenta los juegos de innovación	
Los ejemplos deberían centrarse en detalles técnicos reales.	
Los ejemplos deberían centrarse en los objetivos que se quieren alcanzar.	
Simplifica el diseño y reduce el código innecesario en el desarrollo	

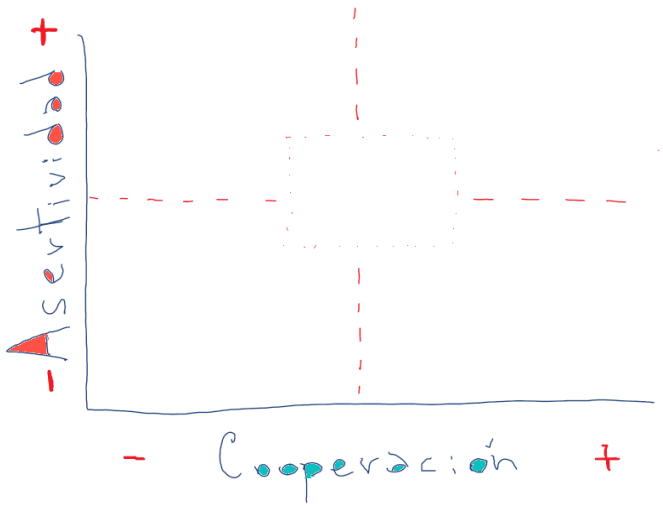
23 Marque los componentes de la inteligencia emocional

Componente	
Actitud	
Autocontrol	
Automotivación	
Capacidad cognitiva	
Conocimiento de uno mismo	
Empatía	
Factor G	
Habilidades sociales	
Inteligencia fluida	

24 Complete el nivel de escucha (1 Interna, 2 Enfocada o 3 Global) de cada uno de los siguientes ítems.

ítem	Nivel
Descubrir el significado subyacente de las palabras ligadas a las emociones de la persona, lo que permite conectarse con la otra persona.	
Escuchar lo que ves, oyes, hueles y sientes de la otra persona. Escuchar todo el entorno de la persona, su lenguaje corporal, su energía y emociones	
Estamos enfocados únicamente en nosotros mismos para reunir información y satisfacer nuestras propias necesidades.	
La atención está en la otra persona para entender sus emociones.	
Mantiene un enfoque en lo que dice la persona, escuchando atentamente cada palabra y "escuchando" cada matiz de la conversación.	
Se profundiza lo que sabe sobre la otra persona al mantener la curiosidad y hacer preguntas aclaratorias durante el diálogo.	
Sirve para recopilar datos, pero no para tener conversaciones profundas.	
Tiende a escuchar más a su propia voz interior, formando opiniones, en lugar de centrarse en lo que se dice	

25 Complete en el gráfico los estilos de conflicto (competir, colaboración, evitar, complaciente, compromiso).



Profesional Ágil – Pablo Lledó ©

26 Complete que tipo de resolución de conflicto recomendado para cada nivel de conflicto. Opciones de resolución: a. Colaboración y consenso; b. Llevar a las partes a una discusión; c. Ceder, negociar y obtener hechos; d. Hablar con cada parte por separado; e. Evitar que las personas se lastimen.

Nivel de Conflicto	Resolución
Contienda	
Cruzada	
Desacuerdo	
Guerra	
Problema por resolver	

27 Marque el tipo de decisión (consulta, consenso, comando) para cada ítem.

ítem	Decisión
La decisión la toma un pequeño grupo de personas	
Los interesados aceptan la decisión tomada por el grupo	
Los miembros del equipo solamente son informados sobre la decisión	
Los participantes se involucran activamente para llegar a una decisión	
Los participantes tienen la oportunidad de brindar aportes	

28 Ordene (1 a 3) los pasos que aplica el líder servicial para empoderar a su equipo.

Ítem	Orden (1 a 3)
Personas	
Proceso	
Propósito	

Profesional Ágil – Pablo Lledó ©

29 Complete verdadero o falso.

En relación con los “métodos de votación” …	V o F
A las personas con el pulgar hacia los lados se los considera en contra de la decisión	
La votación simple podría demorar demasiado tiempo	
La votación simple tiene la oportunidad de escuchar una tercera alternativa	
Votar con los pulgares es más lento que sondear a todos para obtener información	
Votar con un dedo significa que bloqueará la iniciativa	

30 Complete verdadero o falso.

En relación con al “espectro de decisiones de Jim Highsmith” …	V o F
Poder expresar diferencias es suficiente para que las personas se involucren	
Quienes están en contra no apoyarán la iniciativa	
Quienes no estén totalmente a favor deberían buscar otros proyectos	
Utiliza seis escalas	

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

31 El líder del equipo de proyecto está consultando con los miembros del equipo si están de acuerdo o no con el enfoque a utilizar para agregar una nueva funcionalidad a un chatbox de atención al cliente que se encuentra en funcionamiento. ¿Qué herramienta será de utilidad para que los interesados expresen su apoyo, pero al mismo tiempo compartan sus reservas?

A. Voto simple de la mayoría
B. Espectro de decisiones
C. Pulgares arriba / abajo / de lado
D. Disciplined Agile (DA)

32 Los principales interesados de un proyecto sobre mejoras a la atención de clientes están realizando el juego de la caja de productos. ¿Por qué motivo estarán realizando este juego?

A. Estimar en términos relativos la duración de las historias de usuario para priorizar el trabajo pendiente asociado al producto

B. Facilitar modificaciones en el entregable del proyecto para hacer hincapié en la propiedad y responsabilidad de todo el equipo

C. Mejorar el diseño del producto para su mantenimiento sin alterar la funcionalidad esperada

D. Resumir y priorizar la información para vender la idea del proyecto a los otros interesados

33 En un proyecto ágil para el desarrollo de una vacuna de un nuevo virus, las historias de usuario tienen toda la información que necesita el equipo para comenzar con la próxima iteración. Sin embargo, todavía no están claros los criterios para que el cliente pueda utilizar esa vacuna. ¿Qué han definido los interesados de ese proyecto?

A. Definición de listo

B. Definición de terminado

C. Personas

D. Puntos de historia

3.8 Respuesta ejercicios –Interesados

En Scrum un interesado es cualquier persona con interés o influencia sobre el producto que no forma parte del equipo Scrum (dueño del producto, scrum master, equipo de desarrollo). Ejemplo de interesados en Scrum sería el cliente, el patrocinador y los usuarios.

Cliente	√
Dueño del producto	
Equipo de desarrollo	
Patrocinador	√
Scrum master	
Usuario	√

Gestión de interesados

Orden	Proceso
4º	Comunicar periódicamente el estado de avance del proyecto.
2º	Conocer las necesidades de los interesados y los requisitos del proyecto.
5º	Gestionar activamente las expectativas de los interesados.
1º	Identifique periódicamente a los interesados.
3º	Involucrar a los interesados en el proyecto mediante la comunicación activa.

Diagrama del caso de uso. Componentes:

los usuarios
los casos de uso
las interrelaciones

Persona

En relación con una persona en un proyecto ágil…	V o F
Ayudan a guiar las decisiones sobre la funcionalidad y diseño del producto.	V
La persona principal podría quedar satisfecha con una funcionalidad diseñada para otra persona.	F
Podemos inventar personas cuando no tenemos tiempo para una investigación.	F
Representa a un individuo ficticio, que se basa en usuarios reales.	V
Se describe en una ficha su nombre real, su imagen de rostro, sus gustos, etc.	V

No deberíamos "inventar" personas, sino que la persona se descubre como resultado de una investigación para entender cuáles son las necesidades reales del usuario.

La persona principal debe quedar satisfecha con el producto, pero no puede estar satisfecho con una funcionalidad diseñada para otra persona.

Acta de constitución

Componente		
Antecedentes		Al principio del proyecto sabemos poco del producto o servicio final, por lo tanto, tendremos una versión resumida del acta que incluye solamente: Visión, Misión y Criterios de éxito. Cuando el proyecto está más avanzado, se podría agregar información adicional como: Antecedentes, Objetivos, Interesados. No se incluye información adicional como hitos, supuestos y restricciones que suelen formar parte del acta de constitución de proyectos de ciclos predictivos.
Criterios de éxito	√	
Hitos		
Interesados		
Misión	√	
Objetivos		
Restricciones		
Supuestos		
Visión	√	

Caja visión del producto

Componente	
Grupo objetivo	√
Hitos	
Necesidad	√
Objetivos	√
Producto	√
Restricciones	
Supuestos	
Visión	√

Discurso del ascensor (elevator picht): *Para (cliente objetivo), quien (necesita hacer qué) el (producto o servicio) es un (categoría de producto) que (beneficios clave). A diferencia de (competencia), nuestro producto (diferenciación)*.

Herramienta	
Acta de constitución	
Caja de visión	
Discurso del ascensor	√
Esquema de página	
Hoja de ruta del producto	
Product backlog	

Hoja de ruta del producto

La hoja de ruta del producto se estructura en términos de épicas y temas para reflejar la visión de largo plazo del proyecto.

9R Product backlog

En relación con "el listado de trabajo pendiente asociado al producto" …	V o F
El dueño del producto debe involucrar al equipo y a los interesados claves para que den su opinión sobre el producto a desarrollar.	V
Está en constante evolución y permanece disponible para que el equipo y los interesados contribuyan con sugerencias.	V
El equipo de desarrollo y el dueño del producto realizan un refinamiento continuo agregando detalles y estimaciones sobre las características a desarrollar.	V
Las historias de usuario que definan los interesados durante la iniciación se mantendrán estables a lo largo de todo el proyecto.	F

Las historias de usuario que definan los interesados durante la iniciación podrán ir cambiando a lo largo de cada iteración del proyecto, por lo que se podría terminar con un producto diferente a lo que se visualizó en las primeras iteraciones

10R DoD y DoR

Ejemplos	DoD o DoR
Criterios de aceptación y dependencias de la historia de usuario definidos	DoR
El código cumple con los estándares de desarrollo	DoD
Historia de usuario clara, comprobable y factible	DoR
Los cambios de configuración fueron documentados y comunicados	DoD
Persona que aceptará la historia de usuario identificada	DoR
Pruebas unitarias escritas y aprobadas	DoD
Todos los elementos del código tienen que estar completados	DoD

11R Wireframe, Mockup y Prototipo

Ítem	W, M o P
Esqueleto o estructura visual de un producto (ej. página web, app, o productos basados en pantallas).	Wireframe
Esquematiza el diseño de contenidos generales y carece de estilo tipográfico, color o aplicaciones gráficas.	Wireframe
Está muy cerca del producto terminado donde los procesos pueden simularse y la interacción del usuario puede probarse.	Prototipo
Proporciona detalles visuales, como colores y tipografía para dar al cliente una impresión más realista de cómo se verá el producto final.	Mockup
Se enfoca en “qué hace el producto, no cómo se ve”.	Wireframe
Se parece a un producto terminado, pero no es interactivo y no se puede hacer clic en él.	Mockup
Se parece mucho al producto terminado y es una herramienta excelente para obtener comentarios de los usuarios y probar el producto.	Prototipo
Su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de los contenidos.	Wireframe

12R Canales de comunicación.

Canal de comunicación	Interactividad (1 baja a 7 alta)
Audio	3
Cara a cara	7
Carta	1
Chat	5
Email	2
Teléfono	6
Video	4

13R Comunicación osmótica.

Canal de comunicación	
Audio	
Cara a cara	√
Carta	
Chat	
Email	
Teléfono	
Video	

14R Radiador de información.

Radiador de información	
Diagrama de trabajo pendiente	√
Registros de problemas	√
Tablero Kanban	√
Teleconferencia	
Videoconferencia	

15R Diagrama de estacionamiento.

En relación con “el diagrama de estacionamiento” …	V o F
Comunica el estado de avance de productos complejos en los que se trabaja en varias áreas y épicas al mismo tiempo.	V
Hasta que una épica no complete todas las historias de usuario planificadas inicialmente, no podrá considerase como “terminada”.	F
Los cuadros se asemejan a un estacionamiento para autos.	V
Permite a los interesados en el producto realizar preguntas para obtener más información y detectar rápidamente si se está trabajando en la dirección correcta.	V
Presenta el estado de avance de los bloques de funcionalidad de un producto.	V

16R Redes sociales.

En relación con “las redes sociales” …	V o F
En proyectos ágiles, puede reemplaza a las reuniones cara a cara	F
Facilita la distribución de información con equipos distribuidos globalmente	V
Facilitan las comunicaciones de buena calidad	V
Hay que fomentar su uso sin necesidad de protocolos	F
Permite socializar ideas a un bajo costo	V
Pueden reducir el tiempo requerido para las reuniones	V

17R Equipos multicultirales.

En relación con “equipos multiculturales” …	V o F
Capacitar sobre la construcción de equipos e interrelaciones personales para conocer la personalidad de los interesados.	V
Crear una cultura dónde informar problemas no sea bien visto, en su lugar priorizar noticias alentadoras que suban la moral del equipo.	F
Mantener reuniones privadas uno a uno, antes de la reunión formal con todo el equipo.	V
Planificar el horario de las reuniones que vote la mayoría, aunque la minoría tenga que trabajar en horarios anormales.	F

18R Participación de interesados en reuniones.

Tipo de reunión	Participación de interesados externos
Diaria	Baja
Estimación de historias de usuario	Alta
Planificación de la iteración	Media
Priorización del product backlog	Media
Retrospectiva	Baja
Revisión de la iteración	Alta

19R Tormenta de ideas.

Regla	
Comenzar con pocas ideas para facilitar la reunión.	
Proponer ideas descabelladas o impracticables.	√
Sugerir mejoras a las ideas de otros.	√
Todas las ideas son válidas y nadie puede criticarlas al inicio de la sesión.	√

20R Juegos colaborativos.

#	Juego	#	Objetivo
1	Caja de productos	5	Crear empatía con el cliente
2	Mi sombra y Yo	3	Descubrir avances ocultos
3	Bríndeles un jacuzzi	7	Descubrir los peores escenarios ocultos o no considerados
4	Mostrar y decir	1	Identificar las características más interesantes del producto
5	El aprendiz	2	Identificar las necesidades ocultas del cliente
6	Lancha rápida	6	Identificar lo que a los clientes no les gusta de su producto
7	Mi peor pesadilla	4	Identificar los artefactos creados por su producto

21R Complete cada ítem si corresponde a Swarming, Mobbing o Pairing

Ítem	S, M, P
Los miembros del equipo trabajan cada uno con su pc y teclado	Swarming
El equipo se sienta alrededor de una mesa con un solo teclado	Mobbing
Cuando el analista, desarrollador y el probador trabajan en paralelo	Swarming
La salida del monitor va una pantalla grande para que todos puedan ver	Mobbing
Dos miembros del equipo trabajan en la tarea compartiendo un solo teclado	Pairing
Dos programadores cambian los roles de conductor y navegador	Pairing
El conductor es "tácticos", mientras que el observador es "estratégico"	Pairing

22R Especificación mediante ejemplos.

En relación con la “especificación mediante ejemplos (SBE)” …	V o F
Acelera la implementación al acortar los bucles de retroalimentación	V
Alinea las especificaciones con las pruebas de aceptación del usuario	V
Fomenta los juegos de innovación	F
Los ejemplos deberían centrarse en detalles técnicos reales.	F
Los ejemplos deberían centrarse en los objetivos que se quieren alcanzar.	V
Simplifica el diseño y reduce el código innecesario en el desarrollo	V

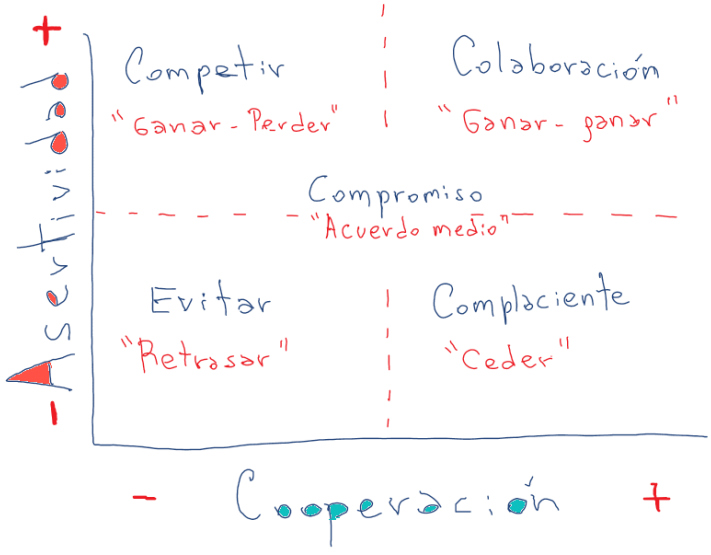
23R Inteligencia emocional

Componente	
Actitud	
Autocontrol	√
Automotivación	√
Capacidad cognitiva	
Conocimiento de uno mismo	√
Empatía	√
Factor G	
Habilidades sociales	√
Inteligencia fluida	

24R Escucha activa.

ítem	Nivel
Descubrir el significado subyacente de las palabras ligadas a las emociones de la persona, lo que permite conectarse con la otra persona.	3
Escuchar lo que ves, oyes, hueles y sientes de la otra persona. Escuchar todo el entorno de la persona, su lenguaje corporal, su energía y emociones	3
Estamos enfocados únicamente en nosotros mismos para reunir información y satisfacer nuestras propias necesidades.	1
La atención está en la otra persona para entender sus emociones.	2
Mantiene un enfoque en lo que dice la persona, escuchando atentamente cada palabra y "escuchando" cada matiz de la conversación.	2
Se profundiza lo que sabe sobre la otra persona al mantener la curiosidad y hacer preguntas aclaratorias durante el diálogo.	2
Sirve para recopilar datos, pero no para tener conversaciones profundas.	1
Tiende a escuchar más a su propia voz interior, formando opiniones, en lugar de centrarse en lo que se dice	1

25R Estilos de conflicto.



Profesional Ágil – Pablo Lledó ©

26R Resolución de conflictos.

Nivel de Conflicto	Resolución
Contienda	Ceder, negociar y obtener hechos.
Cruzada	Hablar con cada parte por separado.
Desacuerdo	Llevar a las partes a una discusión
Guerra	Evitar que las personas se lastimen
Problema por resolver	Colaboración y consenso

27R Tipo de decisión.

Ítem	Decisión
La decisión la toma un pequeño grupo de personas	comando
Los interesados aceptan la decisión tomada por el grupo	consenso
Los miembros del equipo solamente son informados sobre la decisión	comando
Los participantes se involucran activamente para llegar a una decisión	consenso
Los participantes tienen la oportunidad de brindar aportes	consulta

28R Líder servicial

Ítem	Orden (1 a 3)
Personas	2º
Proceso	3º
Propósito	1º

29R Métodos de votación.

En relación con los “métodos de votación” …	V o F
A las personas con el pulgar hacia los lados se los considera en contra de la decisión	F
La votación simple podría demorar demasiado tiempo	F
La votación simple tiene la oportunidad de escuchar una tercera alternativa	F
Votar con los pulgares es más lento que sondear a todos para obtener información	F
Votar con un dedo significa que bloqueará la iniciativa	F

30R Espectro de decisiones.

En relación con al “espectro de decisiones de Jim Highsmith” …	V o F
Poder expresar diferencias es suficiente para que las personas se involucren	V
Quienes están en contra no apoyarán la iniciativa	F
Quienes no estén totalmente a favor deberían buscar otros proyectos	F
Utiliza seis escalas	F

31R Respuesta B. El espectro de decisiones de Jim Highsmith es una herramienta para tomar decisiones donde se les pide a los miembros del equipo que indiquen como se sienten en relación con una decisión: totalmente a favor; de acuerdo, pero con reservas; inseguro con sentimientos opuestos; en contra, pero apoyará la iniciativa; totalmente en contra y vetará la iniciativa. Este es un modelo efectivo porque permite a los interesados indicar su apoyo a una decisión y expresar sus reservas al mismo tiempo.

32R Respuesta D. El juego sobre “La Caja de productos se utiliza para lanzar nuevos proyectos, para facilitar cualquier discusión orientada a la visión y para determinar la gama de un nuevo producto. Se pide a los interesados que imaginen que este producto se venderá en un comercio como cualquier otro producto cotidiano. Cuando los equipos imaginan el producto, en el frente de la caja colocan el nombre del producto, un eslogan, un logotipo y algunos conceptos clave de venta. En la parte posterior de la caja se agrega una descripción detallada, los requisitos y las condiciones de uso. La caja obliga a los equipos a resumir, priorizar y concentrar la información para poder vender la idea a los otros interesados.

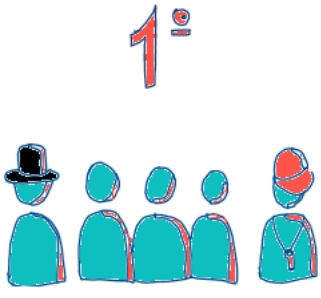
33R Respuesta A. Definición de Listo (DoR: Definition of Ready): las historias de usuario tienen toda la información requerida por el equipo antes de comenzar a trabajar con ellas en una iteración. / Definición de Terminado (DoD: Definition of Done): incluye todos los criterios requeridos para que un producto o lanzamiento sea considerado como terminado para usar por el cliente.

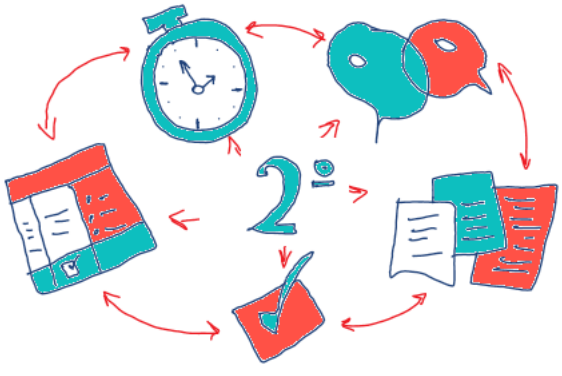
Capítulo 4

Desempeño del equipo

4.1 Roles del equipo

El primer concepto del manifiesto ágil indica que es más importante enfocarse en los individuos y sus interacciones en lugar de los procesos y herramientas. En otras palabras, deberíamos invertir más tiempo en capacitación y desarrollo del equipo, que en mejorar los procesos.





Los procesos y herramientas son mucho más simple de explicar que los temas relacionados con individuos e interacciones, porque los individuos tienen diferentes culturas, habilidades, aptitudes, expectativas, personalidad, etc.

A continuación, comenzaremos presentando los roles básicos de un equipo ágil:

- Dueño del producto
- Equipo de desarrollo
- Líder ágil o Scrum master
- Patrocinador

Dueño del producto

El dueño del producto (product owner) es quien representa los intereses del cliente y del equipo de gestión del valor.

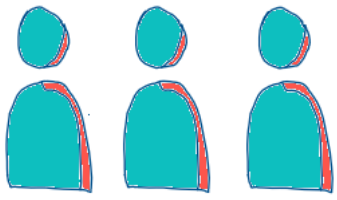
Responsabilidades

- ✓ Definir y priorizar las historias de usuario para maximizar el valor del producto
- ✓ Gestionar el listado de trabajos pendientes (product backlog)
- ✓ Asegurar que el equipo tenga una visión compartida del producto y su valor
- ✓ Proveer los criterios de aceptación del producto al equipo de desarrollo
- ✓ Aceptar, rechazar o solicitar mejoras a cada incremento del producto
- ✓ Gestionar los cambios de funcionalidades y prioridades
- ✓ Gestionar el involucramiento de interesados externos y sus expectativas
- ✓ Proveer la fecha de finalización del proyecto o los lanzamientos
- ✓ Participar de las reuniones de planificación y retrospectiva
- ✓ Último responsable del éxito del desarrollo del producto
- ✓ Gestionar el presupuesto en entornos que no exista un gerente del producto

En grandes organizaciones, además del dueño del producto, existe el rol del **gerente del producto** (product manager) que tiene una responsabilidad ejecutiva sobre los resultados del producto.

El gerente del producto es estratégico, se centra en la visión del producto, los objetivos de la empresa y el mercado. Por otro lado, el dueño del producto es más táctico, traduce la estrategia del gerente del producto en entregables y trabaja con equipos ágiles multifuncionales para asegurarse de que cumplan con los requisitos del proyecto.

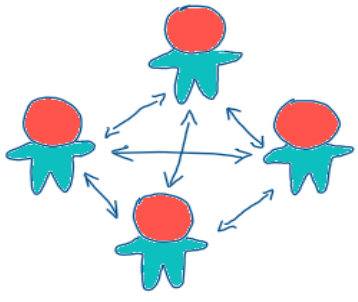
Equipo de desarrollo o entregas



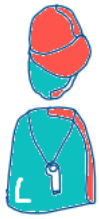
El equipo de desarrollo en ocasiones es llamado Squad (pelotón) al englobar, con una visión más amplia, cualquier habilidad o rol necesario para incluir toda la cadena de valor dentro del equipo. Todos sus miembros están guiados por el mismo propósito, conocen al cliente, el dominio y definen conjuntamente el problema a solucionar.

Responsabilidades

- ✓ Planificar el trabajo de la iteración
- ✓ Autoorganizar y autodirigir sus procesos de trabajo
- ✓ Desarrollar los incrementos del producto
- ✓ Compartir los avances con el equipo en las reuniones diarias
- ✓ Actualizar frecuentemente los radiadores de información
- ✓ Escribir las pruebas de aceptación
- ✓ Realizar control de calidad a cada incremento del producto
- ✓ Demostrar al cliente que el incremento del producto está completo
- ✓ Participar de las reuniones de retrospectiva para mejorar los procesos



Líder del equipo



El líder del equipo o scrum master es un líder servicial del equipo de desarrollo, no el "jefe" o "director" del proyecto. Este líder o facilitador no dirige las tareas de los miembros del equipo, ya que los equipos ágiles se autoorganizan.

Responsabilidades

- ✓ Asegurar que el equipo aplique las prácticas ágiles correctamente
- ✓ Ayudar al equipo de desarrollo a autoorganizarse y autodirigirse
- ✓ Colaborar con el dueño del producto a compartir la visión del producto
- ✓ Colaborar con el dueño del producto a gestionar el product backlog
- ✓ Asegurar la visibilidad del plan y avances del proyecto
- ✓ Proteger al equipo de interferencias, distracciones e interrupciones
- ✓ Eliminar obstáculos para facilitar el desempeño del equipo
- ✓ Facilitar reuniones y eventos productivos
- ✓ Asegurar una buena relación entre el equipo y el dueño del producto
- ✓ Alinear los intereses individuales del equipo con los objetivos del proyecto
- ✓ Mantener reuniones individuales para comprender los intereses personales
- ✓ Proporcionar todos los recursos para la capacitación y desarrollo del equipo
- ✓ Realizar coaching y mentoring al equipo de desarrollo
- ✓ Controlar los desvíos en relación con lo planificado para la iteración
- ✓ Gestionar los conflictos
- ✓ Comunicar, comunicar y comunicar

Coach Ágil: mentor o entrenador con conocimientos y experiencia en metodologías ágiles que puede capacitar, orientar y guiar a organizaciones y equipos a lo largo de su transformación. El líder servicial del equipo podría cumplir este rol de coach, aunque en alguna metodologías como Scrum este suele ser un rol diferente al del scrum master.

Flow Master: coach de un equipo y administrador de solicitudes de servicio que trabaja en un esquema de flujo continuo o en un contexto Kanban; es equivalente al Scrum Master.

Patrocinador

El patrocinador es el principal defensor del proyecto dentro de la organización. Suele ser quien proporciona los recursos y es el portavoz frente a los altos ejecutivos para obtener el apoyo de la organización.

Responsabilidades

- ✓ Autorizar el proyecto
- ✓ Difundir la importancia del proyecto en la organización
- ✓ Defender el proyecto para entregar valor a la organización
- ✓ Comunicar al dueño del producto los objetivos generales del proyecto

El patrocinador no interfiere en el trabajo del equipo, pero podría cancelar el proyecto si no está cumpliendo con sus objetivos.

4.2 Organización del equipo

Un equipo ágil es autoorganizado, responsable y adaptable.

Autoorganizado: dirigido y organizado por sus propios miembros, dentro de los límites preestablecidos por la gerencia para alcanzar los objetivos del proyecto. No existe un único "líder" del equipo durante la vida del proyecto, la persona que dirige el equipo puede cambiar dependiendo de la actividad. La gerencia puede formar al equipo, pero no debería dictar los detalles de cuál es la solución a problemas o cuáles son los procesos para llevar a cabo el trabajo.

Responsable: el equipo se organiza para alcanzar los objetivos y comparte colectivamente la responsabilidad de los resultados. Los miembros del equipo pueden tener habilidades especializadas, pero la responsabilidad pertenece al equipo como un todo, no a una persona individual.

Adaptable: el equipo sabe adaptarse para corregir problemas y mejorar su propio desempeño.

Equipos dedicados

Los equipos ágiles por lo general están 100% dedicados a un solo proyecto y hasta que no completan las actividades asignadas, no comienzan con nuevos proyectos.

Cuando una persona realiza multitareas (multitasking) o cambia frecuentemente entre una tarea y otra (task switch), su productividad es inferior a si estuviera 100% dedicada a una tarea desde el inicio hasta su finalización.

Si una persona está asignada a 2 tareas en simultáneo, debido al costo e ineficiencias de las multitareas, se estima que estará aproximadamente entre un 20% y 40% en cada tarea.[18] Por ejemplo, si pudiera completar 10 puntos de historia por semana dedicado 100% a una única tarea, al estar asignado a 2 tareas en simultáneo (30% en cada una), solamente podrá completar 6 puntos de historia.

[18] PMI Agile Practice Guide.

Especialistas genéricos

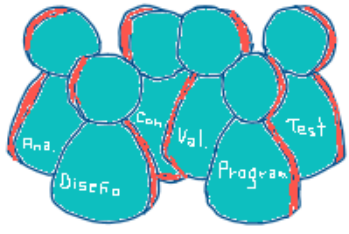
Una de las claves del éxito de los equipos ágiles es trabajar con personas que sean especialistas genéricos (perfil "T" o "peine roto") para que el proyecto no tenga cuellos de botella o problemas por depender de un solo especialista (perfil "I").

Perfil "I": especialistas en un área en particular (gran profundidad), con poca contribución en temas fueras de su especialidad (poca amplitud).

Perfil "T": especialista en un área, pero al mismo tiempo tiene versatilidad y están interesado en colaborar en otras áreas donde tiene menor experiencia. Este ambiente, con personas abiertas e interesadas, donde la comunicación entre todo el equipo es continua, facilita el aprendizaje y la integración de nuevas personas.

Perfil "Peine roto". Imaginemos los dientes de un peine roto con algunos dientes que faltan por completo, algunos cortos y otros más largos. Los dientes del peine representan las diferentes habilidades y el largo de cada diente la profundidad del conocimiento en esa especialidad. Por ejemplo, un diente roto pequeño significa que la persona podría estar desarrollando esa habilidad, mientras que el diente sano completo indica que es un experto en ese tema.

En proyectos ágiles con equipos pequeños, no alcanza con armar un equipo interdisciplinario de perfiles T, sino que deberíamos tener perfiles de "peine roto". De esta forma, si un experto no puede trabajar por el motivo que sea, algún otro miembro del equipo lo podrá reemplazar porque tendrá cierto conocimiento en ese tema y con el tiempo podrá aprender más para especializarse en esa temática.



Trabajo energizante [19]

La práctica de trabajo energizante reconoce que, aunque los profesionales pueden hacer un buen trabajo en circunstancias difíciles, hacen su mejor y más productivo trabajo cuando están energizados y motivados.

Una de las formas más simples de tener energía es cuidarse. Por ejemplo:

- Volver a casa a la misma hora todos los días,
- Compartir tiempo con familiares y amigos,
- Participar de actividades sociales,
- Comer alimentos saludables,
- Hacer ejercicio,
- Dormir bien, etc.

Mientras estemos ocupados con estas actividades fuera de lo laboral, nuestro cerebro se refrescará y seguramente tendremos nuevas ideas la mañana siguiente.

Si el tiempo libre de calidad es el Yin del trabajo energizante, el trabajo enfocado es el Yang. Mientras estemos trabajando, tenemos que enfocarnos en prestar toda la atención, apagar las interrupciones como el correo electrónico y celular, pedirle al líder de proyecto que nos proteja de reuniones innecesarias, etc. Cuando el yin y el yang encajen perfectamente, nos levantaremos por la mañana descansados y ansiosos por comenzar el día. Al final del día, estaremos cansados, pero no agotados y satisfechos con el trabajo realizado.

La programación en pares es una forma de alentar el trabajo energizante porque ayuda a mantener concentrados a los miembros del equipo.

Cuando cometemos más errores que progresos, es hora de tomar un descanso.

Si queremos quedarnos a trabajar horas extras hasta solucionar un problema, suele ser útil que alguien nos recuerde que es hora de ir a descansar. Después de un descanso o una buena noche de sueño, al día siguiente generalmente resolveremos en pocos minutos lo que agotado nos llevaría horas de trabajo.

[19] James Shore, The Art of AgileSM.

Coubicación (Colocación)

Según los principios del manifiesto ágil el método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara. Este principio dio lugar a que las metodologías ágiles recomienden colocar a todos los miembros de su equipo en un mismo lugar físico.

El equipo coubicado o colocado está ubicado en el mismo lugar físico para poder comunicarse y colaborar cara a cara. La coubicación favorece el soporte proactivo, discusiones libres, colaboración abierta, compartir el conocimiento tácito y la comunicación osmótica.

El espacio de trabajo debería contar con equipamiento y herramientas tales como:

- Espacio visible para colocar los radiadores de información
- Pizarras, rotafolios, post-its, papeles adhesivos
- Mesa redonda con proyector y pantalla grande
- Tecnología para videoconferencias grupales
- Café, alimentos, juguetes de entretenimiento, etc.

Reuniones de espacio abierto (Open Space Meetings): es un estilo de "desconferencia" autoorganizada donde los participantes crean y administran su propia agenda de sesiones paralelas en torno a un tema específico. Las personas proponen sesiones que les gustaría liderar, o sesiones que les gustaría ver a otros exponer. Rápidamente surge un horario, compuesto de sesiones sobre temas que las personas sienten pasión por seguir. Si un participante se encuentra en una reunión donde no puede aprender o contribuir, se trasladará a otra sala.

Cabe destacar que desde el año 2001 que se redactó el manifiesto ágil hasta el día de la fecha, el mundo cambió. En especial a partir del año 2020 donde el Covid-19 y las regulaciones de distanciamiento social hicieron que la forma preferida de trabajo sea con los miembros del equipo trabajando de manera remota desde sus hogares. Cuando la colocación en un mismo espacio físico sea imposible, se puede realizar una colocación virtual utilizando tecnología como videoconferencias, sharepoints, chats, etc.

Equipos distribuidos (virtuales)

Cuando los equipos están distribuidos en diferentes lugares, es necesario un esfuerzo adicional para asegurar que las comunicaciones fluyan sin mayores complicaciones.

Trabajar con equipos virtuales presenta desventajas o dificultades para poder:

- Comprender el lenguaje corporal, cambios de voz, percibir olores, etc.
- Coordinar reuniones cuando las personas viven en diferentes usos horarios.
- Construir una relación más cercana. Por ejemplo, ir al bar después del trabajo.

Si bien gestionar equipos ubicados en diferentes lugares físicos suele ser más complicado que cuando están colocados, hay varias ventajas de los equipos virtuales tales como:

- Menores gastos de oficina
- Mayor acceso a empleados que viven en cualquier lugar del mundo
- Mayor acceso a personas con diferencias culturales que facilitan la creatividad
- Mayor flexibilidad para trabajar en diferentes horarios
- Ahorro en costos y tiempo de traslados a la oficina, mejorando la calidad de vida al poder dedicar ese tiempo a la familia, amigos, ejercicio, arte, diversión, etc.

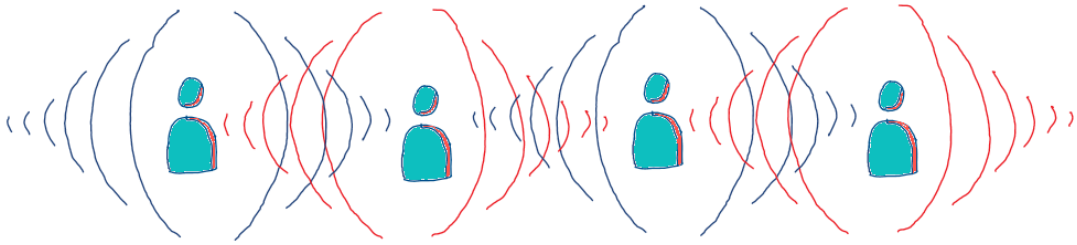
Para gestionar equipos virtuales es necesario invertir en tecnologías para facilitar las comunicaciones. Por ejemplo: salas virtuales con ventanas de pecera, videoconferencias, chat, pizarras electrónicas interactivas, software, sharepoints, etc.

Ventanas de pecera (fishbowl windows): establecer enlaces de videoconferencia duraderos entre los diversos lugares en los que el equipo está disperso. Los miembros del equipo inician el enlace al principio de un día de trabajo y lo cierran al final del día. De esta manera, el equipo puede verse y conectarse espontáneamente, reduciendo el retraso en la colaboración inherente a la separación geográfica.

Aun en entornos de equipos virtuales, si fuera posible, organice de vez en cuando reuniones presenciales para facilitar la construcción de relaciones y mejorar la confianza entre los miembros del equipo.

Comunicación osmótica

La comunicación osmótica ocurre cuando las personas trabajan tan cerca entre sí, sin barreras, que pueden escuchar las conversaciones de los demás. La información fluye en el fondo auditivo de los miembros del equipo, de tal forma que ellos captan la información relevante, como si fuera por ósmosis.



Este tipo de comunicación sucede normalmente al estar colocados en la misma sala física y en menor escala cuando están colocados en una sala virtual. Así, cuando una persona hace una pregunta, los demás pueden estar atentos para contribuir al debate si fuera necesario; o dejar de prestar atención y continuar con su trabajo cuando el tema no sea relevante.

Para mejorar la comunicación osmótica es necesario que los miembros del equipo trabajen lo más cerca entre sí posible, eliminando cualquier tipo de barreras en la comunicación.

La comunicación osmótica hace que el costo de las comunicaciones sea bajo y la tasa de retroalimentación alta, de modo que rápidamente se corrigen los errores y se difunde el conocimiento.

Desventajas:

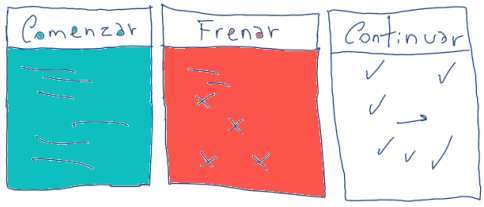
- Ambiente ruidoso que dificulta la concentración
- Aquellos expertos que tienen que interrumpir su trabajo permanentemente para responder preguntas, tienen una menor productividad
- Falta de privacidad para proteger temas personales o confidenciales

4.3 Desarrollo del equipo

Los equipos ágiles examinan su desempeño y buscan formas de mejorar de manera continua, permaneciendo abiertos a la mentoría, coaching y capacitación. Sin embargo, no responden bien al control de la alta gerencia, ya que prefieren resolver las cosas por su cuenta con autonomía.

El momento para la reflexión del equipo es durante la retrospectiva al finalizar cada iteración, donde se comparte lo que salió bien y mal para mejorar el proceso de la próxima iteración. En la reunión de retrospectiva se contestan tres preguntas:

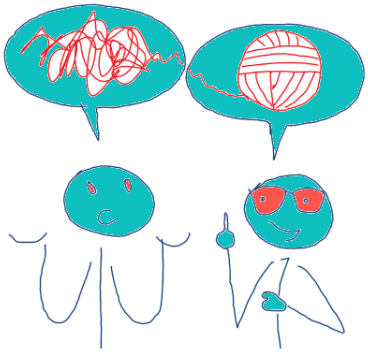
- ¿Qué deberíamos comenzar a hacer?
- ¿Qué deberíamos dejar de hacer?
- ¿Qué deberíamos seguir haciendo?



Los equipos de alto rendimiento toman en serio esta reflexión y ajustan su comportamiento en consecuencia. Con el tiempo, los equipos desarrollan mejores prácticas que contribuyen con la productividad y velocidad del equipo.

La mentoría y el coaching son herramientas de gran utilidad para el desarrollo del equipo.

- Mentoría: el mentor comparte con los miembros del equipo sus experiencias, habilidades, conocimientos y consejos para guiarlos en su desarrollo profesional.
- Coaching: el coach no comparte su punto de vista, sino que fomenta el autoconocimiento de una persona para que tome las mejores decisiones por si mismo.

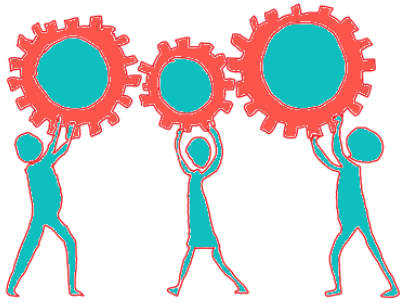


Equipos de alto rendimiento

Un equipo ágil de alto rendimiento es un pequeño grupo de personas, máximo 12 personas, con habilidades complementarias que están comprometidas con un propósito común y se responsabilizan mutuamente por el éxito o fracaso del proyecto.

Características del equipo: [20]

- Autoorganizados
- Empoderados para tomar decisiones
- Capaces de resolver cualquier problema
- Comprometidos con el éxito del equipo
- Dueños de sus decisiones y compromisos
- Motivados por la confianza
- Orientados por el consenso
- Están en un constante desacuerdo constructivo



Equipos de alto rendimiento	vs	Equipos de bajo rendimiento
Tienen objetivos claros y realistas		
Se basan en la confianza		Ausencia de confianza
Gestionan bien los conflictos		Miedo al conflicto
Comprometidos y autoorganizados		Falta de compromiso
Sentido de pertenencia		Evitar rendición de cuentas
Toman decisiones por consenso		Falta atención a resultados
Dedicados a tiempo completo		

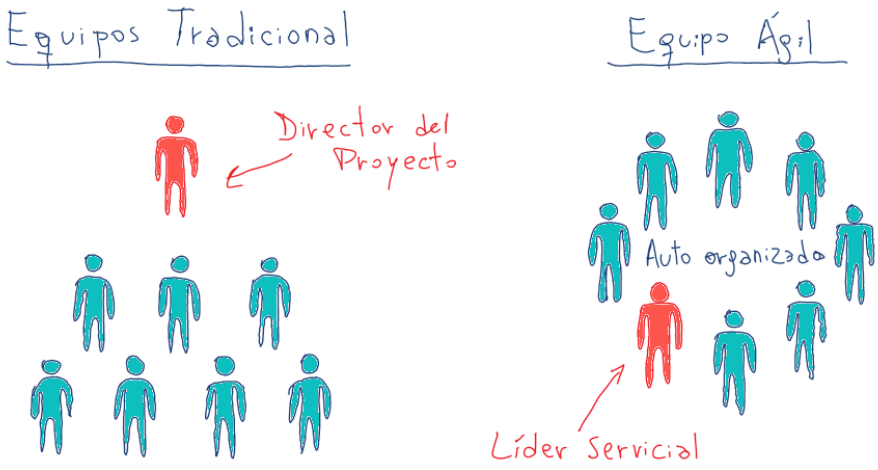
Los equipos de alto rendimiento no siempre llegarán a un rápido consenso sobre como afrontar un problema, por el contrario, tendrán discusiones divergentes que fomenten la creatividad y alternativas frente a un mismo problema, antes de llegar a un acuerdo convergente.

[20] Lyssa Adkins, Coaching Agile Teams.

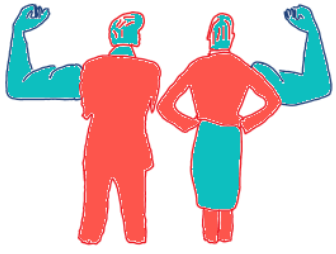
Empoderar al equipo

En la gestión tradicional de proyectos el director de proyectos podría ser la persona con mayor autoridad o jerarquía del equipo para poder ejercer, cuando sea necesario, una gestión de comando y control sobre los miembros de su equipo.

Por el contrario, los equipos ágiles necesitan ser empoderados para autogestionarse y autoorganizarse para alcanzar los objetivos del proyecto. El líder servicial crea un ambiente de confianza entre los miembros del equipo para mantener altos niveles de motivación.



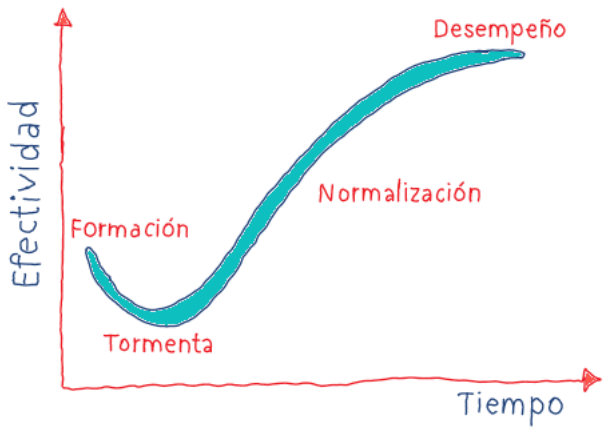
Características de los equipos ágiles empoderados



- Autoorganizados y autodirigidos
- Capacidad para tomar decisiones autónomas
- Trabajo y decisiones en equipo
- Responsabilidad y propiedad sobre las actividades
- Motivación y orgullo por lo que están haciendo
- Estructuras livianas con mínima supervisión
- Respeto y comprensión mutua
- Entusiasmo para trabajar

Escalera de Tuckman [21]

Bruce Tuckman define fases por las que atraviesan la mayoría de los equipos antes de llegar al alto rendimiento: 1º Formación, 2º Tormenta, 3º Normalización, 4º Desempeño.

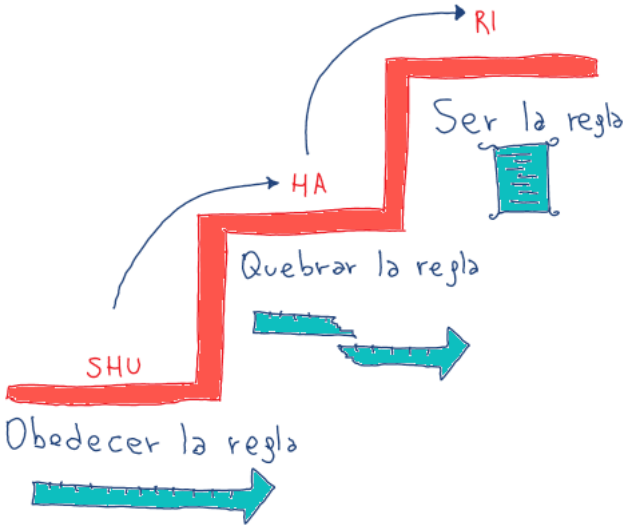


Formación	Se adquiere el equipo y se establecen las reglas básicas. Los miembros del equipo son tratados como extraños.	El equipo evolucionará más rápido si el objetivo del proyecto y los roles del equipo se expresan claramente.
Tormenta	Los individuos comienzan a comunicar sus sentimientos, pero no se sienten parte del equipo. Se resisten al control de los líderes del grupo y muestran hostilidad.	El líder debe promover las buenas relaciones y una actitud positiva para minimizar los conflictos.
Normalización	Las personas se sienten parte del equipo y reconocen que pueden llevar a cabo su trabajo si aceptan otros puntos de vista.	El líder ayuda al equipo a que encuentren soluciones por sí mismos.
Desempeño	El equipo trabaja en un ambiente abierto con gran confianza, donde la flexibilidad es la clave del éxito y las jerarquías no tienen importancia.	El equipo está funcionando al máximo y los objetivos se alcanzan con eficiencia. El equipo toma posesión colectiva y los actos de culpar y acusar han desaparecido.

[21] Bruce Tuckman, Desarrollo secuencial en grupos pequeños.

Shu-Ha-Ri

"Shu-Ha-Ri" se traduce como "Obedecer-Desprenderse-Superarse". Esta filosofía permite al equipo entender y adaptar metodologías ágiles a sus propias necesidades como grupo.



Shu

Obedecer las reglas. Aprender técnicas fundamentales que se aplican directamente sin modificaciones. Por ejemplo, contestar solamente las 3 preguntas básicas de las reuniones diarias: 1. ¿Qué hiciste ayer?, ¿Que harás hoy? y ¿Hay impedimentos en tu camino? Recomendado para los equipos con poca experiencia.

Ha

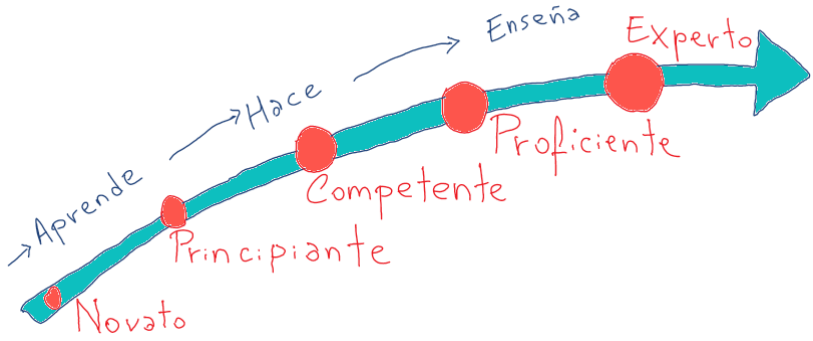
Quebrar las reglas. Se permite un poco más de flexibilidad para despegarse conscientemente de las reglas en temas creativos para adaptar las metodologías. Por ejemplo: agregar nuevas preguntas a las 3 básicas. Recomendado para equipos proficientes.

Ri

Ser la regla. Se libera al equipo de toda estructura guiadora, de modo que experimenten con lo aprendido, basados en las necesidades del equipo para desarrollar nuevas reglas o metodologías. Por ejemplo, crear preguntas diferentes a las recomendadas por la metodología. Solamente utilizado por algunos equipos expertos.

Modelo de Dreyfus

El modelo Dreyfus de adquisición de competencias indica las fases que atraviesan las personas al adquirir habilidades a través de la instrucción formal y la práctica: 1º. Novato, 2º. Principiante, 3º Competente, 4º Proficiente y 5º Experto.



Novato: tiene una adherencia rígida a las reglas o planes sin aplicar su juicio discrecional.

Principiante: tiene una percepción situacional limitada y trata todos los aspectos del trabajo con igual importancia relativa.

Competente: puede acumular bastante información y trabajar en múltiples actividades; tiene cierta percepción de sus acciones en relación con los objetivos; planifica deliberadamente y formula rutinas.

Proficiente: tiene una visión holística del problema, prioriza los aspectos importantes, percibe desviaciones de lo normal y se adapta a cada situación.

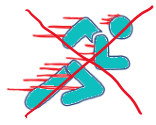
Experto: sabe lo que es posible; se mueve por la intuición y no necesita reglas o principios; utiliza un enfoque analítico en situaciones nuevas.

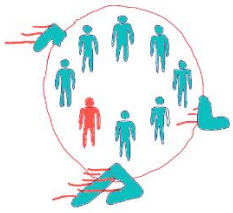
	Nivel de Habilidad				
	Novato	Principiante	Competente	Proficiente	Experto
Adquirir conocimientos	No situacional	Situacional	Situacional	Situacional	Situacional
Reconocimiento	Descompuesto	Descompuesto	Holístico	Holístico	Holístico
Toma decisiones	Analítica	Analítica	Analítica	Intuitiva	Intuitiva
Conciencia	Monitoreo	Monitoreo	Monitoreo	Monitoreo	Absorción

4.4 Seguimiento del equipo

El enfoque ágil para monitorear el progreso del proyecto suele estar en la velocidad y desempeño del equipo.

La velocidad individual (ej. historias de usuario completadas) como métrica de desempeño de un miembro del equipo es inapropiada. Si un miembro del equipo sabe que están midiendo su velocidad individual, podría no dedicar tiempo a ayudar a otros miembros de su equipo.

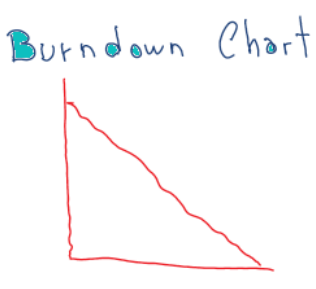




Las personas deberían estar motivadas a trabajar en equipo para aprender cosas nuevas y colaborar para mejorar la productividad. En caso de que un equipo tenga un rendimiento bajo, el equipo debe tratar de descubrir y explorar las posibles formas de mejora como equipo y también como individuos.

Por su parte, en proyectos de desarrollo de software, contar la cantidad de líneas de código (SLOC: source lines of code) que realiza un equipo no sirve para analizar el desempeño o productividad del equipo. Por ejemplo, un software con pocas líneas de código podría haber demorado más tiempo y ser de mejor calidad que otro software con más líneas de código.

A continuación, presentaremos diferentes gráficas que son utilizadas como radiadores de información para analizar la velocidad del equipo y para estimar las fechas de finalización del proyecto o de un lanzamiento.



Gráfica de Trabajo Pendiente (Burndown Chart)

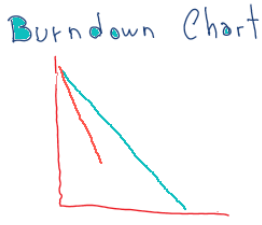
El gráfico de trabajo pendiente (Burndown Chart o Sprint Burndown) representa el trabajo pendiente de finalización frente al tiempo restante.

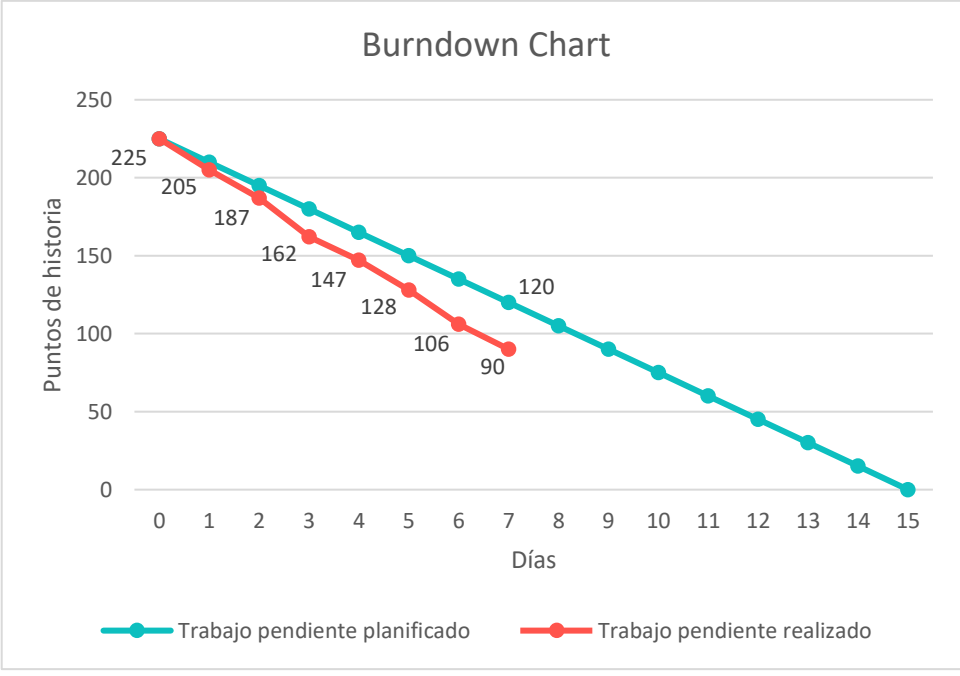
Este gráfico es un radiador de información utilizado para mostrar el progreso de la iteración a través del tiempo y para estimar el rendimiento diario del equipo.

En la tabla a continuación se presenta un ejemplo de un proyecto que ha planificado completar 225 puntos de historia en los próximos 15 días.

Días	Puntos de historia planificados	Trabajo pendiente Planificado	Trabajo realizado	Trabajo acumulado realizado	Trabajo pendiente Real
0	0	225	0	0	225
1	15	210	20	20	205
2	15	195	18	38	187
3	15	180	25	63	162
4	15	165	15	78	147
5	15	150	19	97	128
6	15	135	22	119	106
7	15	120	16	135	90
8	15	105			
9	15	90			
10	15	75			
11	15	60			
12	15	45			
13	15	30			
14	15	15			
15	15	0			

El gráfico de trabajo pendiente compara los datos del trabajo pendiente planificado con el trabajo pendiente realizado a través del tiempo.

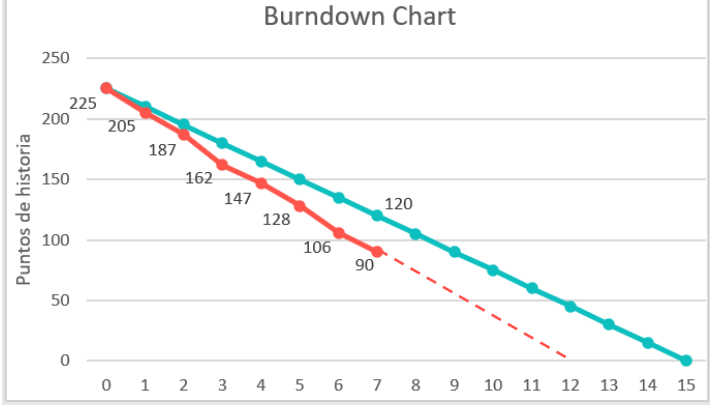




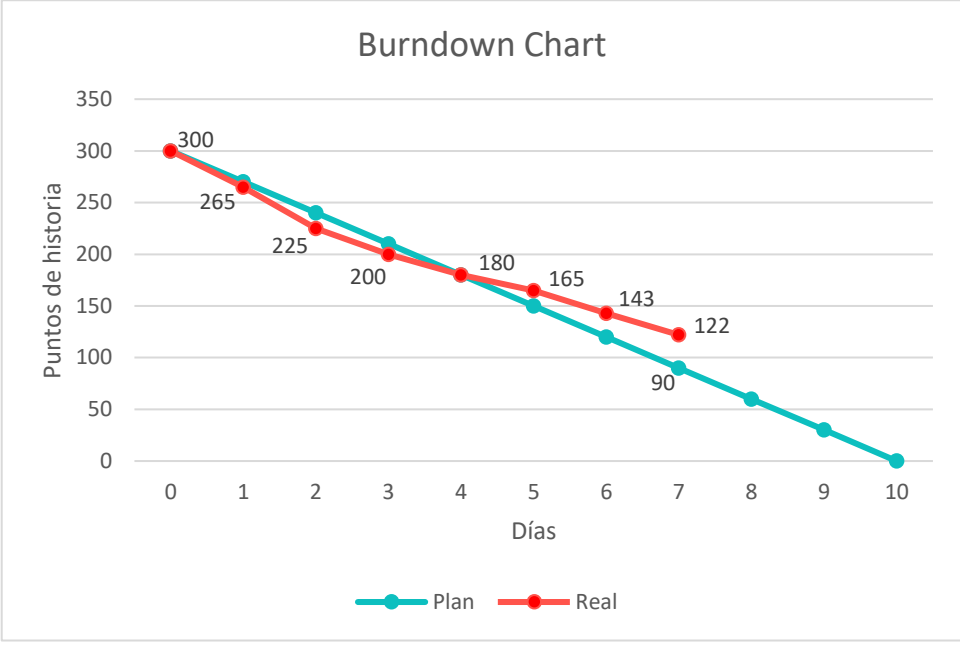
Cuando la línea de trabajo pendiente realizado está por debajo del trabajo pendiente planificado, significa que el equipo está trabajando más rápido de lo estimado.

En este ejemplo el equipo estimó trabajar a un ritmo promedio de 15 puntos de historia por día y está trabajando a un promedio de 19 puntos de historia por día, por lo que al final del día 7 queda un trabajo pendiente de 90 puntos de historia, en lugar de los 120 puntos de historia que deberían estar pendientes a esa fecha según el plan original.

Si esta velocidad se mantiene, se puede estimar que el proyecto finalizará en el día 12 (225 puntos de historia / 19 velocidad promedio), o sea, 3 días antes de lo planificado.

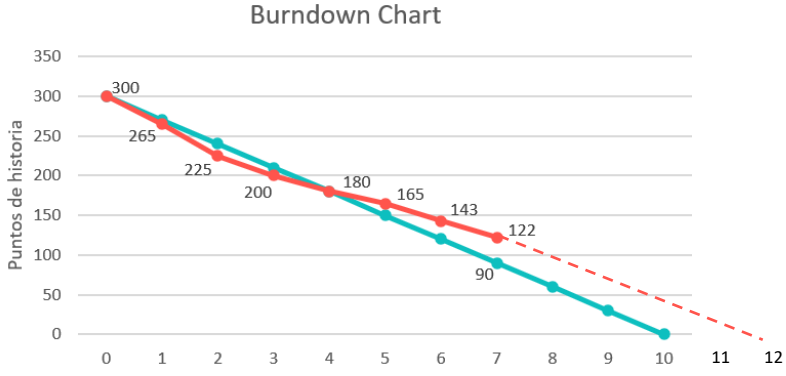


Por su parte, cuando la línea de trabajo pendiente realizado está por encima del trabajo pendiente planificado, el proyecto está más lento de lo estimado.



En este otro ejemplo, el equipo estimó trabajar a un ritmo promedio de 30 puntos de historia por día (300 puntos totales entre 10 días). Sin embargo, el equipo está trabajando a un promedio de 25 puntos de historia por día, por lo que al final del día 7 queda un trabajo pendiente de 122 puntos de historia, en lugar de los 90 puntos de historia que deberían estar pendientes a esa fecha según el plan original.

Por lo tanto, al final del día 7, el proyecto está retrasado en 32 puntos de historia (122 – 90). Si se mantuviera una velocidad promedio de 25, el proyecto finalizará en el día 12 (300 puntos de historia / 25 velocidad), o sea 2 días más tarde de lo planificado.

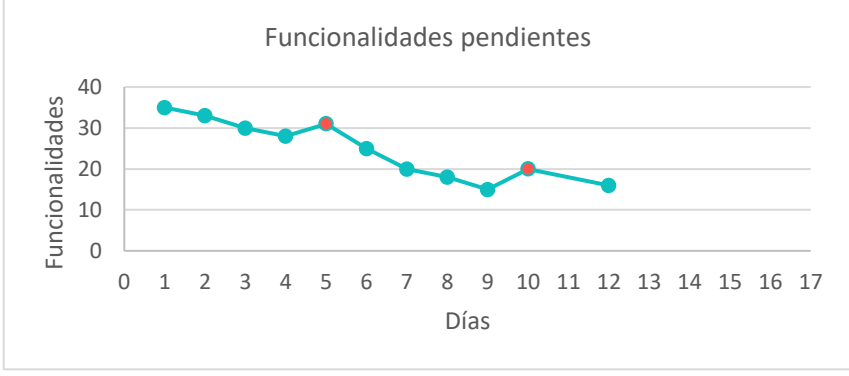


En la tabla a continuación se muestran los datos del gráfico anterior, donde se puede observar que en los días 1 y 2 el trabajo realizado fue superior al trabajo planificado y a partir del día 3 se trabaja menos que lo planificado.

Días	Trabajo planificado	Trabajo pendiente según plan	Trabajo realizado	Trabajo acumulado realizado	Trabajo pendiente real
0	0	300	0	0	300
1	30	270	35	35	265
2	30	240	40	75	225
3	30	210	25	100	200
4	30	180	20	120	180
5	30	150	15	135	165
6	30	120	22	157	143
7	30	90	21	178	122
8	30	60			
9	30	30			
10	30	0			
Total	**300**		**178**		
Trabajo realizado promedio = 178 / 7 = 25,4					

Diagrama de funcionalidades pendientes (Burndown feature chart)

Este diagrama es similar al anterior, pero en lugar de representar los puntos de historia pendientes, se grafican las funcionalidades pendientes a través del tiempo.



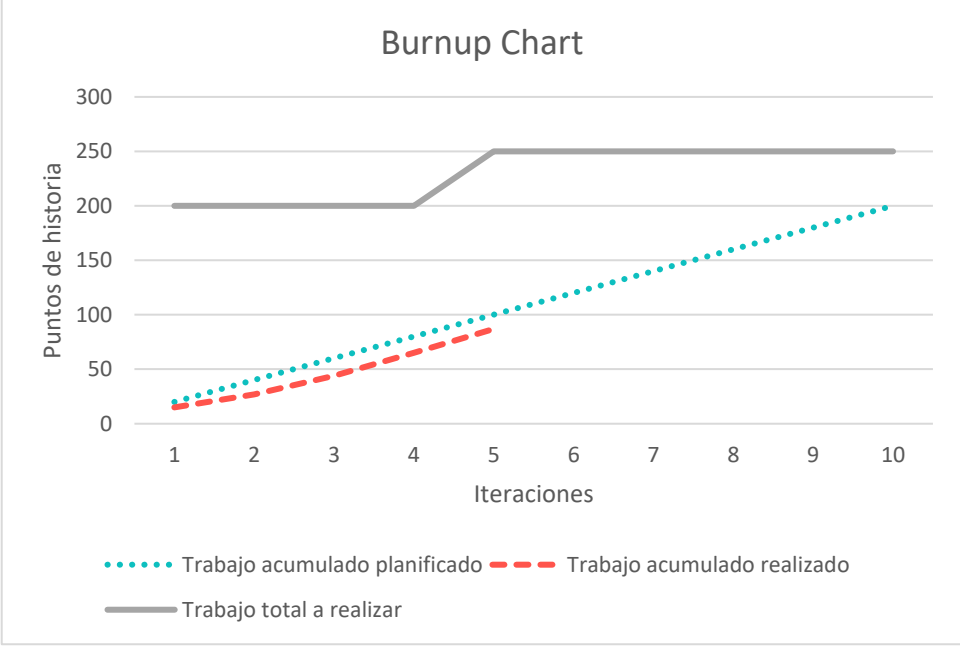
Si la línea descendente de funcionalidades a través del tiempo indica entre un día y otro un incremento (ejemplo: entre el día 4 y 5, o entre el 9 y 10), seguramente se han agregado nuevas funcionalidades al proyecto.

Gráfica de Trabajo Realizado (Burnup Chart)

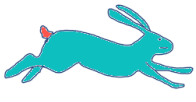
La gráfica de Trabajo Realizado es una representación gráfica del trabajo completado acumulado a través del tiempo.

Suele incluir una línea horizontal con el total del trabajo a realizar, que varía hacia abajo o hacia arriba cuando hay cambios en el alcance del proyecto. Esta barra horizontal facilita identificar rápidamente los cambios de alcance en relación con la gráfica de trabajo pendiente.

En el gráfico a continuación se presenta un ejemplo donde el alcance aumentó en 50 puntos de historia al comenzar con la iteración 5.

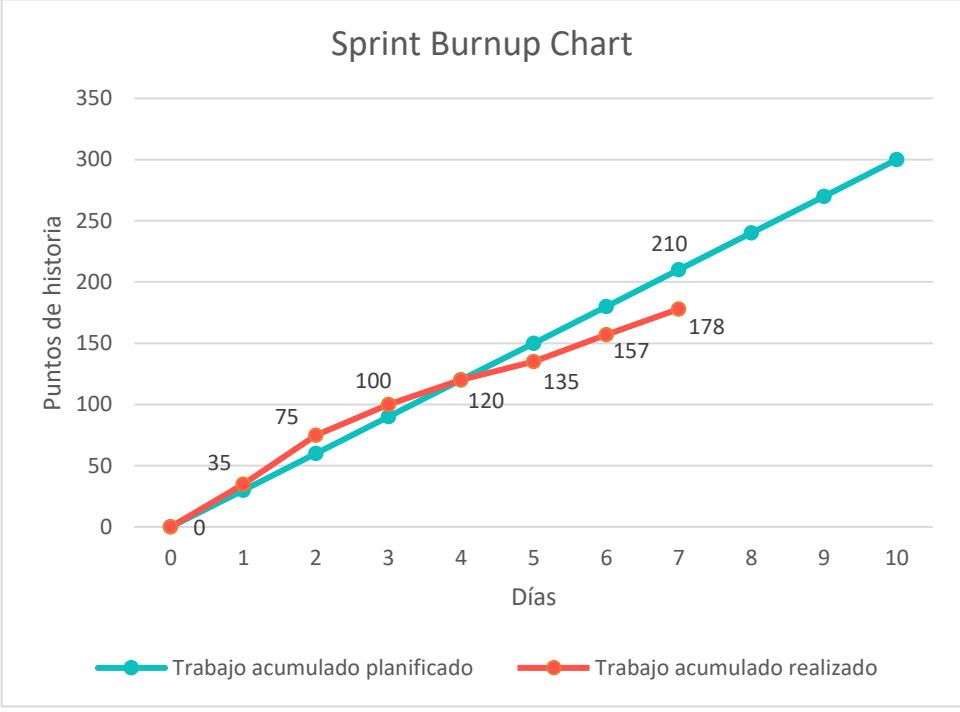


Cuando la línea del trabajo acumulado realizado está por debajo del trabajo acumulado planificado, el proyecto está lento.



Por otro lado, si el trabajo realizado está por encima de lo planificado, el proyecto va más rápido de lo planificado.

En el próximo ejemplo, al finalizar el día 7 se ha completado un trabajo acumulado de 178 puntos de historia. Sin embargo, según el plan a esa fecha se deberían haber completado 210 puntos de historia. Por lo tanto, el proyecto está lento en 32 puntos de historia.



Por otro lado, 178 puntos de historia se deberían haber completado aproximadamente en el día 6 según el plan; por lo que podemos decir que al finalizar el día 7 el proyecto tiene un retraso de 1 día.

También se puede estimar el SPI (Schedule performance index) comparando las historias completadas vs las historias planificadas.

SPI = Historias completadas / Historias planificadas = 178 / 210 = 0,85. Por lo tanto, el equipo está trabajando a un 85% de la tasa prevista. Si el SPI es menor que 1 indica que el proyecto está lento y cuando sea mayor que 1 indica que el proyecto va más rápido que lo planificado.

Una forma de estimar la fecha de finalización del proyecto suponiendo que se mantendrá la misma ineficiencia que hasta la fecha sería con esta regla de 3 simple:

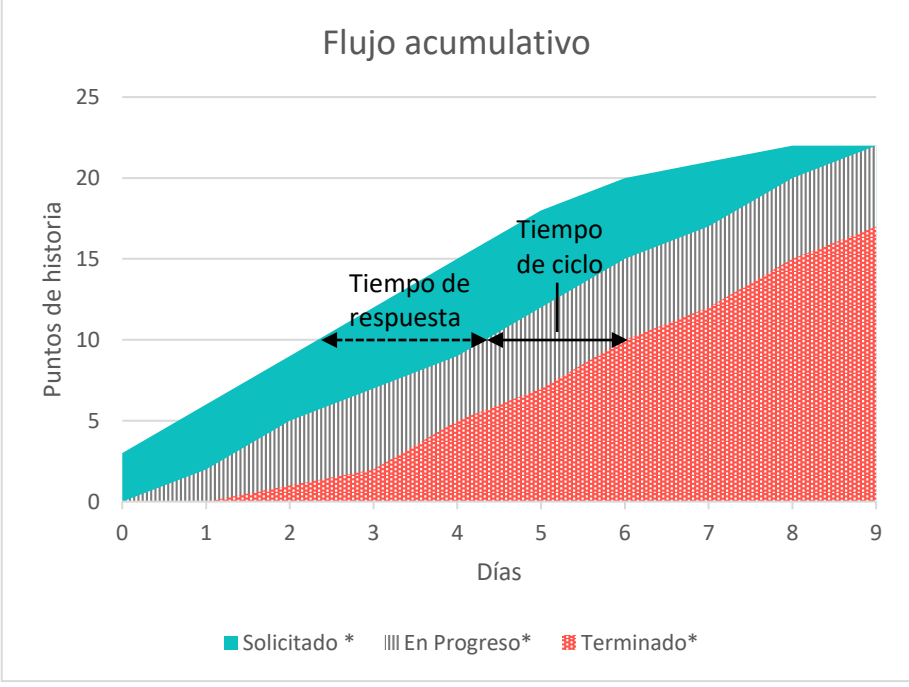
178 puntos de historia _____ 7 días

300 puntos de historia _____ X días

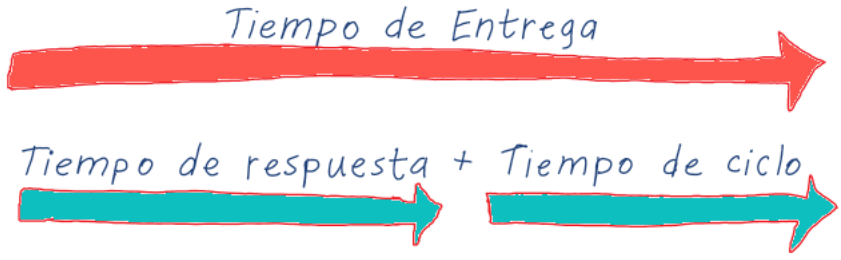
X = (300 x 7) / 178 = 11,8 días

Diagrama de flujo acumulativo

Si al Burnup Chart que representa las tareas completadas, le agregamos información de las tareas que están en progreso y de las tareas que no han comenzado, obtenemos la representación de un diagrama de flujo acumulativo como se presenta en el ejemplo a continuación.



Como explicamos en el Capítulo 2, la diferencia horizontal del trabajo solicitado acumulado está indicando el tiempo de respuesta. Mientras que la diferencia horizontal del trabajo en progreso acumulado indica el tiempo de ciclo. La sumatoria del tiempo de respuesta más el tiempo de ciclo es el tiempo de entrega.

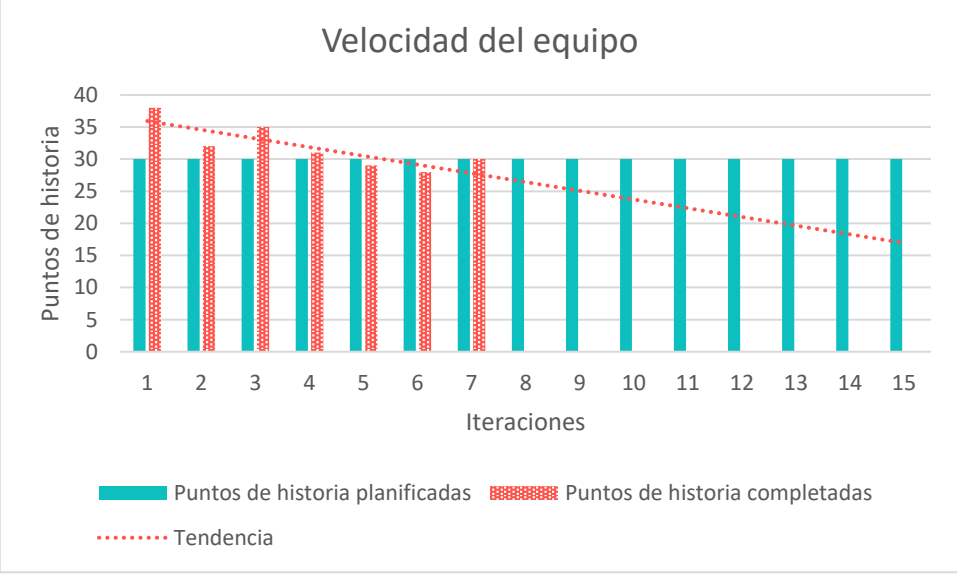


Velocidad del equipo

La velocidad es una medida de la cantidad de trabajo que un equipo puede realizar durante una iteración.

La velocidad se calcula al final de la iteración sumando la cantidad de trabajo realizado. Por ejemplo, si al finalizar una iteración se completaron 3 historias de usuario de 5 puntos de historia cada una, la velocidad de esa iteración fue de 15 puntos de historia.

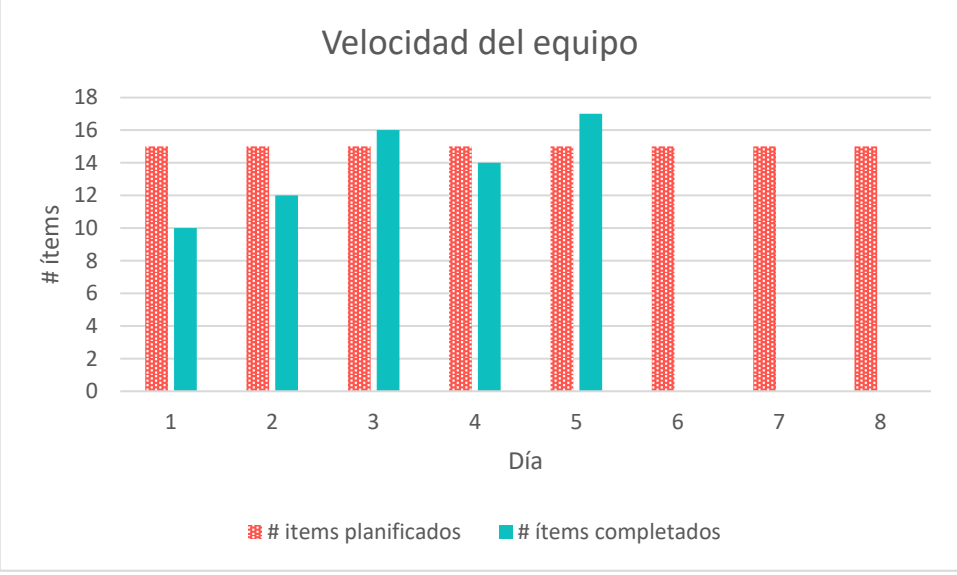
La velocidad del equipo se puede representar en el **gráfico de velocidad** que indica la cantidad de trabajo completado por el equipo en cada iteración.



El gráfico indica cierta variación en las primeras iteraciones, lo cuál es normal en las fases iniciales del proyecto, seguidas de una mayor estabilidad en las siguientes iteraciones. Por lo tanto, en este ejemplo los datos de velocidad son normales.

No deberíamos prestar atención a la tendencia decreciente en este ejemplo ya que hay muy pocos datos para poder estimar con precisión una tendencia. Además, se observa que en las últimas 4 iteraciones el equipo ya está trabajando a una velocidad promedio similar a lo planificado (29,5 puntos de historia en promedio), por lo que no hay datos que indiquen que vaya a disminuir la velocidad en las futuras iteraciones.

La velocidad se debe medir y seguir en la misma unidad de medida que utilizó el equipo para estimar el trabajo. Por ejemplo, si estimaron número de ítems a realizar por día, la velocidad se medirá por número de ítems completados cada día. Si las dos unidades fueran diferentes, el equipo no le podrá hacer seguimiento a su progreso en comparación con sus estimaciones.



Para poder estimar con números de ítems, es necesario planificar en base a datos históricos de ítems de tamaño similar en otros proyectos realizados en el pasado.

En la mayoría de los proyectos ágiles dónde no existen datos históricos de proyectos similares y las historias de usuario son de diferentes tamaños relativos, se suelen utilizar “puntos de historia” para estimar el trabajo y medir la velocidad.

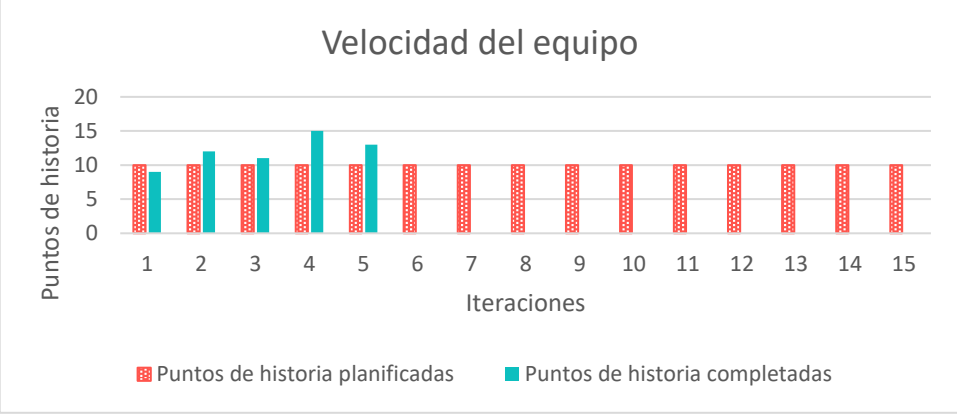
Por lo general, la velocidad suele incrementar gradualmente en las primeras iteraciones cuando el equipo va mejorando su desarrollo y luego se estabiliza a medida que el producto se va tornando más complejo (más bugs o errores, más documentación, más dependencias, etc.)

Los puntos de historia y velocidad de un equipo no son comparables con los de otros equipos. Cada equipo asigna puntos de historia a las tareas que van a desarrollar en función de la complejidad relativa de sus historias de usuario. Las historias de usuario de cada equipo son diferentes, por lo que la estimación relativa de los puntos de historia también será diferente. Por lo tanto, no se puede analizar la velocidad o eficiencia de cada equipo comparando puntos de historia finalizados por diferentes equipos.

Estimaciones

La velocidad del equipo sirve para predecir la cantidad de trabajo que el equipo puede hacer en futuras iteraciones.

Ejemplo 1. Si para la próxima iteración de 15 días se estimó una velocidad promedio de 10 puntos de historia por día y al finalizar el día 5 el equipo completó 60 puntos de historia, podemos estimar que el proyecto está adelantado. Según el plan en el día 6 se deberían haber completado 60 puntos de historia (6 días x 10 puntos de historia). Sin embargo, los 60 puntos de historia se completaron el día 5, por lo que podemos decir que el proyecto está adelantado en 1 día según el plan original.



Ejemplo 2. Los puntos de historia estimados para el lanzamiento del proyecto son 420. Luego de 8 iteraciones de una semana cada una, se han completado 275 historias de usuario con una velocidad de trabajo bastante estable. ¿Cuántas iteraciones faltan para completar el proyecto?

Historias restantes: 420 – 275 = 145.

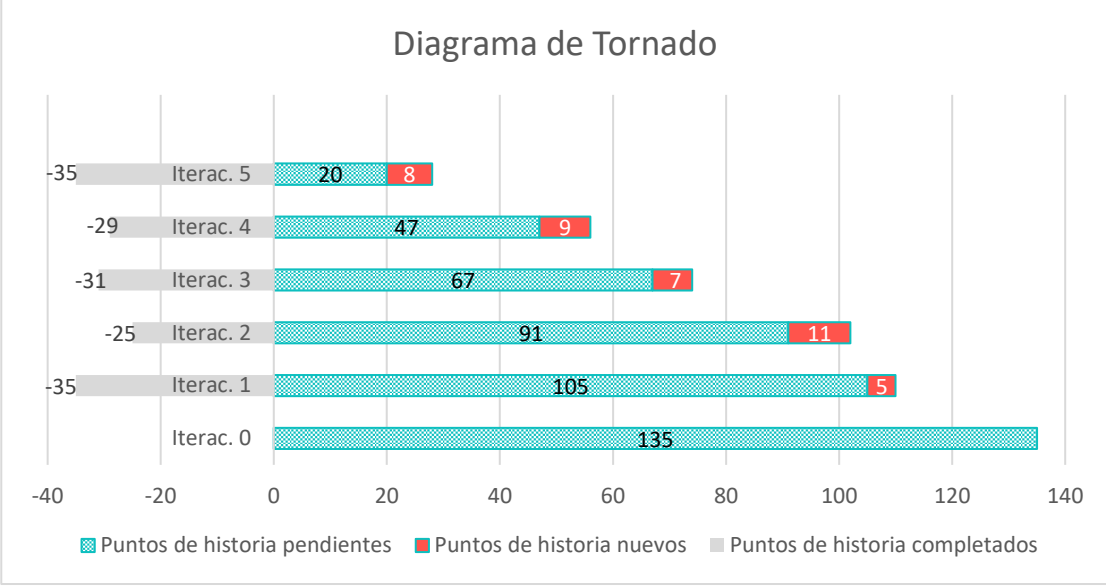
Velocidad promedio por iteración: 275 / 8 = 34,38.

Iteraciones restantes para completar el proyecto = Historias restantes / Velocidad promedio = 145 / 34,38 = 4,22.

El proyecto va a terminar en la iteración 5. Aunque el redondeo de 4,22 es 4 iteraciones, se debe redondear siempre hacia arriba, ya que el proyecto seguirá incompleto al finalizar la iteración 4.

Diagrama de Tornado (Hurricane-style charts)

Las gráficas de velocidad no representan los cambios en el alcance del proyecto. Para visualizar los cambios de alcance se puede utilizar un diagrama de Tornado como el ejemplo que se presenta a continuación.



Como podemos observar en todas las iteraciones se ha estado agregando alcance adicional representado en el gráfico como “puntos de historia nuevos”.

¿En qué iteración se estima que podría terminar este proyecto?

Podemos suponer que el proyecto tendrá a futuro un comportamiento similar al del pasado. O sea, el equipo seguirá con un desempeño promedio de 31 puntos de historia por iteración (promedio simple de las historias completadas en cada iteración: 35, 25, 31, 29, 35).

Por otro lado, suponemos que las nuevas funcionalidades que se agregan al proyecto serán similares a las del pasado con 8 puntos de historia promedio por iteración (promedio simple de los nuevos puntos de historia: 5, 11, 7, 9, 8).

Bajo este escenario de un trabajo pendiente de 20, nuevas funcionalidades que se podrían agregar de 8 y una velocidad promedio de 31, el proyecto debería finalizar en la próxima iteración # 6.

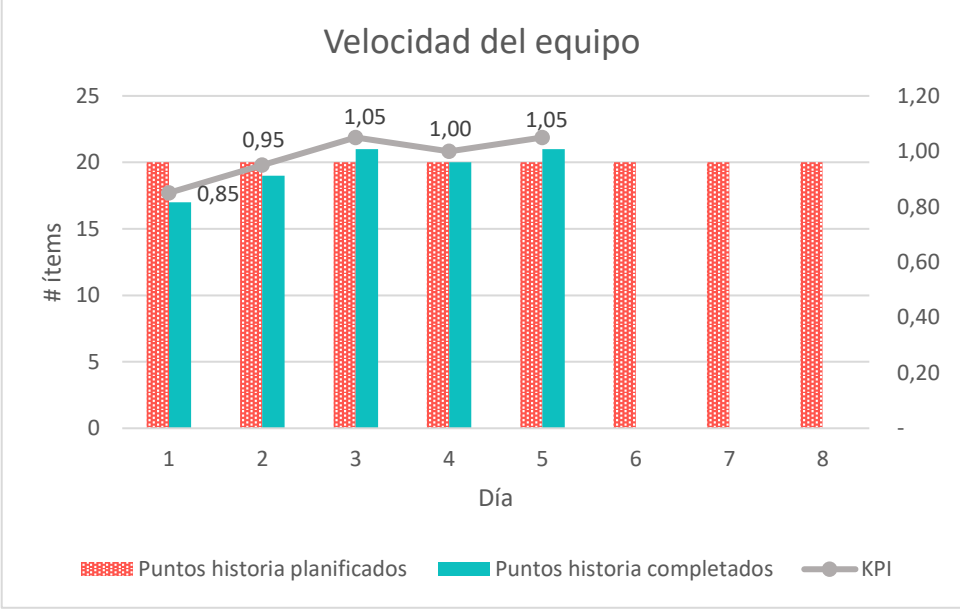
Compromiso del equipo

Los indicadores clave de rendimiento (KPI: key performance indicator) indican el desempeño del proyecto a lo largo del tiempo para evaluar los planes actuales y desarrollar los planes futuros.

Algunos ejemplos de KPI ágiles son: puntos de historia completados por iteración, defectos detectados por iteración, cantidad de deuda técnica, etc.

Un KPI para medir el compromiso de los miembros del equipo sería mediante la relación entre los puntos de historia completados vs los puntos de historia planificados.

Compromiso = historias completadas / historias planificadas



En el ejemplo se observa que en la iteración 1 el KPI de 0,85 indica que los miembros del equipo estuvieron un 15% por debajo del rendimiento planificado. En el otro extremo, en la iteración 5, el KPI de 1,05 indica que el equipo está con un desempeño del 5% superior a lo planificado.

El KPI promedio de las 5 iteraciones está en 0,98, lo que indica que el equipo está un 2% por debajo del objetivo planificado.

4.5 Ejercicios – Equipo

1 Marque a quién corresponde principalmente cada una de las siguientes responsabilidades: Product owner (PO); Equipo de desarrollo (E); Scrum master (SM); Patrocinador (P).

Responsabilidad	PO	E	SM	P
Aceptar, rechazar o solicitar mejoras a cada incremento del producto				
Actualizar frecuentemente los radiadores de información				
Alinear los intereses individuales con los objetivos del proyecto				
Asegurar la visibilidad del plan y avances del proyecto				
Asegurar que el equipo aplique las prácticas ágiles correctamente				
Asegurar que el equipo tenga una visión compartida del producto				
Autorizar el inicio del proyecto				
Ayudar al equipo de desarrollo a autoorganizarse y autodirigirse				
Cancelar el proyecto si no está cumpliendo con sus objetivos				
Comunicar al dueño del producto los objetivos generales del proyecto				
Definir y priorizar las historias de usuario				
Demostrar al cliente que el incremento del producto está completo				
Desarrollar los incrementos del producto				
Difundir la importancia del proyecto en la organización				
Escribir las pruebas de aceptación				
Gestionar el listado de trabajos pendientes y sus cambios				
Gestionar los conflictos				
Planificar el trabajo de la iteración				
Proporcionar los recursos para la capacitación y desarrollo del equipo				
Proteger al equipo de interferencias, distracciones e interrupciones				
Proveer la fecha de finalización del proyecto o los lanzamientos				
Proveer los criterios de aceptación del producto al equipo de desarrollo				
Realizar coaching y mentoring al equipo				

2 Indique cuál de las siguientes frases corresponden al producto owner (PO) o al gerente del producto (GP).

Frase	PO	GP
Asegura que cumplan los requisitos del proyecto		
Más estratégico		
Más táctico		
Se centra en la visión del producto, los objetivos de la empresa y el mercado		
Tiene una responsabilidad ejecutiva sobre los resultados del producto		
Traduce la estrategia en entregables		

3 Marque verdadero (V) o falso (F).

Frase	V	F
El coach ágil es un mentor o entrenador con experiencia en metodologías ágiles		
El coach ágil guía a organizaciones y equipos a lo largo de su transformación		
El flow master administra las solicitudes de servicio en un contexto Kanban		
El flow master sería equivalente al scrum master		
El líder servicial del equipo siempre cumple el rol de coach ágil		
En Scrum el rol de coach ágil lo cumple el scrum master		

4 Con relación a proyectos ágiles, marque verdadero (V) o falso (F).

Frase	V	F
Cambiar frecuentemente entre una tarea y otra mejora la creatividad		
El equipo comparte colectivamente la responsabilidad de los resultados		
El equipo está dedicado a un solo proyecto		
El proyecto es dirigido y organizado por sus propios miembros		
Existe un único “líder” del equipo durante la vida del proyecto		
La gerencia dicta los detalles de los procesos para llevar a cabo el trabajo		
Trabajar en 2 tareas a la vez, implica dedicar entre un 20%-40% a cada tarea		

5 Marque el perfil de especialistas (I, T, Peine roto (PR) de las siguientes frases.

Frase	I	T	PR
El proyecto está sufriendo cuellos de botella al depender de un especialista			
Especialista en un área, pero al mismo tiempo tiene versatilidad			
Especialistas en un área en particular (gran profundidad)			
Experto en cierto temas y desarrollando otras especialidades			
Interesado en colaborar en otras áreas donde tiene menor experiencia			
Poca contribución en temas fueras de su especialidad (poca amplitud)			

6 Marque cuál de las siguientes acciones están relacionadas con el trabajo energizante.

Acción	√
Comer alimentos saludables	
Compartir tiempo con familiares y amigos	
Dormir bien	
Hacer ejercicio	
Multitareas	
Participar de actividades sociales	
Participar de todas las reuniones	
Programación en pares	
Trabajar horas extras para solucionar problemas	
Volver a casa a horario todos los días	

7 Marque verdadero (V) o falso (F).

Frase	V	F
El lugar de trabajo requiere un espacio visible para los radiadores de información		
En una reunión de espacio abierto, si un participante no puede aprender o contribuir se trasladará a otra sala.		
Las ventanas de pecera son salas multiuso que facilitan la coubicación del equipo		
Los equipos distribuidos en diferentes usos horarios generan reuniones más eficientes		
Los equipos distribuidos virtualmente favorecen el conocimiento tácito		
Los equipos distribuidos virtualmente favorecen la comunicación osmótica		

8 Marque las características que corresponden a los equipos de alto rendimiento

Característica	√
Deciden de manera individual para acelerar el proyecto	
Enfrentan los conflictos	
Evitan la rendición de cuentas	
Evitan los conflictos	
Llegan siempre a un rápido consenso	
Se basan en la confianza	
Se basan en los procesos	
Tienen discusiones divergentes	
Toman decisiones por consenso	

9 Marque verdadero (V) o falso (F).

Frase	V	F
El líder ágil ejerce comando y control sobre los miembros de su equipo		
El scrum master es la persona con mayor autoridad del equipo		
Los equipos ágiles son empoderados		
Los equipos empoderados tienen estructuras livianas con mínima supervisión		

10 Marque a qué fase de la escalera de Tuckman (Formación, Tormenta, Normalización, Desempeño) corresponde cada una de las siguentes frases.

Frase	Fase
El equipo evoluciona más rápido si sus roles se expresan claramente	
El equipo toma posesión colectiva y no se culpa a nadie	
El equipo trabaja en un ambiente abierto con gran confianza	
El líder ayuda al equipo a que encuentren soluciones por sí mismos	
El líder promueve las buenas relaciones para minimizar los conflictos	
Las jerarquías no tienen importancia	
Las personas no se sienten parte del equipo	
Los individuos se resisten al control de los líderes	
Los miembros del equipo aceptan otros puntos de vista	
Los miembros del equipo son tratados como extraños	
Se establecen las reglas básicas	

11 Marque a qué parte de Shu-Ha-Ri corresponde cada una de las siguentes frases.

Frase	Shu	Ha	Ri
Aprender técnicas fundamentales que se aplican sin modificaciones			
Desarrollar nuevas reglas o metodologías			
Despegarse de las reglas para adaptar las metodologías			
Obedecer las reglas			
Quebrar las reglas			
Se libera al equipo de toda estructura guiadora			
Ser la regla			

12 Marque a qué fase (novato, principiante, competente, proficiente, experto) del modelo de Dreyfus corresponden las siguientes frases.

Frase	Fase
Acumula bastante información y trabaja en múltiples actividades	
Adherencia rígida a las reglas o planes sin aplicar su juicio discrecional	
Planifica deliberadamente y formula rutinas	
Prioriza los aspectos importantes y percibe desviaciones de lo normal	
Recolección de información no situacional	
Sabe lo que es posible	
Se mueve por la intuición y no necesita reglas o principios	
Tiene cierta percepción de sus acciones en relación con los objetivos	
Tiene una visión holística del problema	
Trata todos los aspectos del trabajo con igual importancia relativa	

13 En relación Burndown chart, marque verdadero (V) o falso (F).

Frase	V	F
Compara los datos del trabajo pendiente planificado con el trabajo realizado acumulado a través del tiempo		
Es de gran utilidad para visualizar los cambios de alcance del proyecto		
Es un radiador de información		
Representa el trabajo pendiente de finalización frente al tiempo restante		
Se puede utilizar para graficar las funcionalidades pendientes a través del tiempo		
Si la línea de trabajo pendiente realizado está por encima del trabajo pendiente planificado, significa que el equipo está trabajando más rápido de lo estimado.		
Sirve para proyectar la fecha de finalización del proyecto		

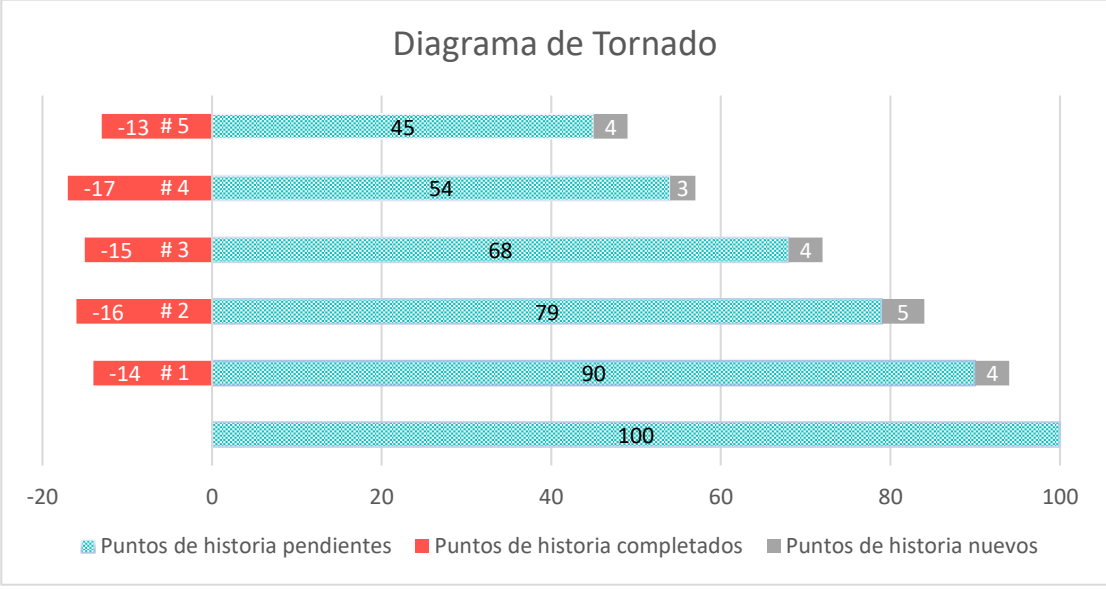
14 En relación Burnup chart, marque verdadero (V) o falso (F).

Frase	V	F
Cuando la línea del trabajo acumulado realizado está por encima del trabajo acumulado planificado, el proyecto está lento		
El SPI se calcula dividiendo las historias completadas sobre las historias planificadas		
Facilita identificar los cambios de alcance		
Representa el trabajo completado acumulado a través del tiempo		
Si el SPI es menor que 1 indica que el proyecto está rápido		
Si el trabajo realizado está por encima de lo planificado, el proyecto aumentó el alcance		
Suele incluir una línea horizontal con el total del trabajo total a realizar		

15 En relación con la velocidad del equipo, marque verdadero (V) o falso (F).

Frase	V	F
El compromiso del equipo se puede calcular comparando los puntos de historia completados vs los planificados		
Indica la cantidad de trabajo pendiente al finalizar cada iteración		
La velocidad de un equipo se puede comparar con la velocidad de otros equipos		
Mide la cantidad de trabajo que un equipo puede realizar durante una iteración		
Se calcula en medio de la iteración		
Se puede utilizar como unidad de medida “puntos de historia” de cada iteración		
Se puede utilizar como unidad de medida el número de ítems a realizar por día		
Sirve para predecir la cantidad de trabajo a realizar en futuras iteraciones		
Suele disminuir gradualmente en las primeras iteraciones		
Suele tener variación en las primeras iteraciones		

16 Según los datos del diagrama a continuación de las últimas 5 iteraciones del proyecto, ¿En qué iteración se estima que terminará el proyecto?



Respuesta: _____

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

17 Durante una reunión de capacitación sobre técnicas para el desarrollo del equipo ágil, uno de los participantes que no está conforme con un expositor, se levanta y se va a otra sala. Esto podría ser un ejemplo de ____

A. Falta de cumplimiento del acta de equipo
B. Equipo en fase de tormenta
C. Reunión de retrospectiva
D. Reuniones de espacio abierto

18 Usted es el coach de un equipo ágil que está desarrollando un sistema de facenote para seguridad informática. Los miembros del equipo, mientras trabajan con acoso (mobbing), están atravesando discusiones técnicas divergentes de cómo avanzar con el desarrollo. ¿Qué debería hacer en esta situación?

A. Intervenir en la discusión para ayudar al equipo a encontrar una convergencia
B. Observar atentamente las discusiones y dejar que el equipo encuentre una solución
C. Recomendar al equipo que comiencen a trabajar en pares
D. Invitar al cliente y patrocinador para que participen de esa discusión

19 ¿En cuál de las siguientes herramientas se visualiza mejor un cambio de alcance en un proyecto ágil?

A. Gráfica de trabajo realizado
B. Gráfica de trabajo pendiente
C. Gráfica de velocidad del equipo
D. Diagrama de Gantt

4.6 Respuesta ejercicios – Equipo

Responsabilidades del equipo

Responsabilidad	PO	E	SM	P
Aceptar, rechazar o solicitar mejoras a cada incremento del producto	√			
Actualizar frecuentemente los radiadores de información		√		
Alinear los intereses individuales con los objetivos del proyecto			√	
Asegurar la visibilidad del plan y avances del proyecto			√	
Asegurar que el equipo aplique las prácticas ágiles correctamente			√	
Asegurar que el equipo tenga una visión compartida del producto	√			
Autorizar el inicio del proyecto				√
Ayudar al equipo de desarrollo a autoorganizarse y autodirigirse			√	
Cancelar el proyecto si no está cumpliendo con sus objetivos				√
Comunicar al dueño del producto los objetivos generales del proyecto				√
Definir y priorizar las historias de usuario	√			
Demostrar al cliente que el incremento del producto está completo		√		
Desarrollar los incrementos del producto		√		
Difundir la importancia del proyecto en la organización				√
Escribir las pruebas de aceptación		√		
Gestionar el listado de trabajos pendientes y sus cambios	√			
Gestionar los conflictos			√	
Planificar el trabajo de la iteración		√		
Proporcionar los recursos para la capacitación y desarrollo del equipo			√	
Proteger al equipo de interferencias, distracciones e interrupciones			√	
Proveer la fecha de finalización del proyecto o los lanzamientos	√			
Proveer los criterios de aceptación del producto al equipo de desarrollo	√			
Realizar coaching y mentoring al equipo			√	

2R Product owner y Gerente del producto

Frase	PO	GP
Asegura que cumplan los requisitos del proyecto	√	
Más estratégico		√
Más táctico	√	
Se centra en la visión del producto, los objetivos de la empresa y el mercado		√
Tiene una responsabilidad ejecutiva sobre los resultados del producto		√
Traduce la estrategia en entregables	√	

3R Coach ágil

Frase	V	F
El coach ágil es un mentor o entrenador con experiencia en metodologías ágiles	√	
El coach ágil guía a organizaciones y equipos a lo largo de su transformación	√	
El flow master administra las solicitudes de servicio en un contexto Kanban	√	
El flow master sería equivalente al scrum master	√	
El líder servicial del equipo siempre cumple el rol de coach ágil		√
En Scrum el rol de coach ágil lo cumple el scrum master		√

4R Organización del equipo

Frase	V	F
Cambiar frecuentemente entre una tarea y otra mejora la creatividad		√
El equipo comparte colectivamente la responsabilidad de los resultados	√	
El equipo está dedicado a un solo proyecto	√	
El proyecto es dirigido y organizado por sus propios miembros	√	
Existe un único “líder” del equipo durante la vida del proyecto		√
La gerencia dicta los detalles de los procesos para llevar a cabo el trabajo		√
Trabajar en 2 tareas a la vez, implica dedicar entre un 20%-40% a cada tarea	√	

5R Tipos de especialistas.

Frase	I	T	PR
El proyecto está sufriendo cuellos de botella al depender de un especialista	√		
Especialista en un área, pero al mismo tiempo tiene versatilidad		√	
Especialistas en un área en particular (gran profundidad)	√		
Experto en cierto temas y desarrollando otras especialidades			√
Interesado en colaborar en otras áreas donde tiene menor experiencia		√	
Poca contribución en temas fueras de su especialidad (poca amplitud)	√		

6R Trabajo energizante

Acción	
Comer alimentos saludables	√
Compartir tiempo con familiares y amigos	√
Dormir bien	√
Hacer ejercicio	√
Multitareas	
Participar de actividades sociales	√
Participar de todas las reuniones	
Programación en pares	√
Trabajar horas extras para solucionar problemas	
Volver a casa a horario todos los días	√

7R Ubicación del equipo

Frase	V	F
El lugar de trabajo requiere un espacio visible para los radiadores de información	√	
En una reunión de espacio abierto, si un participante no puede aprender o contribuir se trasladará a otra sala.	√	
Las ventanas de pecera son salas multiuso que facilitan la coubicación del equipo		√
Los equipos distribuidos en diferentes usos horarios generan reuniones más eficientes		√
Los equipos distribuidos virtualmente favorecen el conocimiento tácito		√
Los equipos distribuidos virtualmente favorecen la comunicación osmótica		√

8R Equipos de alto rendimiento

Característica	
Deciden de manera individual para acelerar el proyecto	
Enfrentan los conflictos	√
Evitan la rendición de cuentas	
Evitan los conflictos	
Llegan siempre a un rápido consenso	
Se basan en la confianza	√
Se basan en los procesos	
Tienen discusiones divergentes	√
Toman decisiones por consenso	√

9R Equipo empoderado

Frase	V	F
El líder ágil ejerce comando y control sobre los miembros de su equipo		√
El scrum master es la persona con mayor autoridad del equipo		√
Los equipos ágiles son empoderados	√	
Los equipos empoderados tienen estructuras livianas con mínima supervisión	√	

10R Tuckman

Frase	Fase
El equipo evoluciona más rápido si sus roles se expresan claramente	Formación
El equipo toma posesión colectiva y no se culpa a nadie	Desempeño
El equipo trabaja en un ambiente abierto con gran confianza	Desempeño
El líder ayuda al equipo a que encuentren soluciones por sí mismos	Normalización
El líder promueve las buenas relaciones para minimizar los conflictos	Tormenta
Las jerarquías no tienen importancia	Desempeño
Las personas no se sienten parte del equipo	Tormenta
Los individuos se resisten al control de los líderes	Tormenta
Los miembros del equipo aceptan otros puntos de vista	Normalización
Los miembros del equipo son tratados como extraños	Formación
Se establecen las reglas básicas	Formación

11R Shu-Ha-Ri

Frase	Shu	Ha	Ri
Aprender técnicas fundamentales que se aplican sin modificaciones	√		
Desarrollar nuevas reglas o metodologías			√
Despegarse de las reglas para adaptar las metodologías		√	
Obedecer las reglas	√		
Quebrar las reglas		√	
Se libera al equipo de toda estructura guiadora			√
Ser la regla			√

12R Dreyfus

Frase	Fase
Acumula bastante información y trabaja en múltiples actividades	Competente
Adherencia rígida a las reglas o planes sin aplicar su juicio discrecional	Novato
Planifica deliberadamente y formula rutinas	Competente
Prioriza los aspectos importantes y percibe desviaciones de lo normal	Proficiente
Recolección de información no situacional	Novato
Sabe lo que es posible	Experto
Se mueve por la intuición y no necesita reglas o principios	Experto
Tiene cierta percepción de sus acciones en relación con los objetivos	Competente
Tiene una visión holística del problema	Proficiente
Trata todos los aspectos del trabajo con igual importancia relativa	Principiante

13R Burndown chart

Frase	V	F
Compara los datos del trabajo pendiente planificado con el trabajo realizado acumulado a través del tiempo		√
Es de gran utilidad para visualizar los cambios de alcance del proyecto		√
Es un radiador de información	√	
Representa el trabajo pendiente de finalización frente al tiempo restante	√	
Se puede utilizar para graficar las funcionalidades pendientes a través del tiempo	√	
Si la línea de trabajo pendiente realizado está por encima del trabajo pendiente planificado, significa que el equipo está trabajando más rápido de lo estimado.		√
Sirve para proyectar la fecha de finalización del proyecto	√	

14R Burnup chart

Frase	V	F
Cuando la línea del trabajo acumulado realizado está por encima del trabajo acumulado planificado, el proyecto está lento		√
El SPI se calcula dividiendo las historias completadas sobre las historias planificadas	√	
Facilita identificar los cambios de alcance	√	
Representa el trabajo completado acumulado a través del tiempo	√	
Si el SPI es menor que 1 indica que el proyecto está rápido		√
Si el trabajo realizado está por encima de lo planificado, el proyecto aumentó el alcance		√
Suele incluir una línea horizontal con el total del trabajo total a realizar	√	

15R Velocidad del equipo

Frase	V	F
El compromiso del equipo se puede calcular comparando los puntos de historia completados vs los planificados	√	
Indica la cantidad de trabajo pendiente al finalizar cada iteración		√
La velocidad de un equipo se puede comparar con la velocidad de otros equipos		√
Mide la cantidad de trabajo que un equipo puede realizar durante una iteración	√	
Se calcula en medio de la iteración		√
Se puede utilizar como unidad de medida “puntos de historia” de cada iteración	√	
Se puede utilizar como unidad de medida el número de ítems a realizar por día	√	
Sirve para predecir la cantidad de trabajo a realizar en futuras iteraciones	√	
Suele disminuir gradualmente en las primeras iteraciones		√
Suele tener variación en las primeras iteraciones	√	

16R Gráfico de Tornado

Iteración	Puntos de historia pendientes	Puntos de historia completados	Puntos de historia nuevos
# 0	100		
# 1	90	-14	4
# 2	79	-16	5
# 3	68	-15	4
# 4	54	-17	3
# 5	45	-13	4
	Promedio	-15	4

Promedio de puntos de historia completados por iteración: 15

Promedio de puntos de historia nuevos por iteración: 4

Puntos de historia completados netos del nuevo alcance: 15 – 4 = 11

Puntos de historia pendientes en la última iteración: 45

Iteraciones restantes = Pendiente / Completados netos = 45 / 11 = 4,09 (5 redondeando hacia arriba).

Por lo tanto, suponiendo una velocidad y nuevo alcance similar al promedio histórico, se estima que quedan 5 iteraciones adicionales para finalizar el proyecto.

Respuesta: El proyecto terminará en la Iteración #10

Demostración:

Iteración	Puntos de historia pendientes	Puntos de historia completados	Puntos de historia nuevos
# 0	100		
# 1	90	-14	4
# 2	79	-16	5
# 3	68	-15	4
# 4	54	-17	3
# 5	45	-13	4
# 6	34	-15	4
# 7	23	-15	4
# 8	12	-15	4
# 9	1	-15	4
# 10	0	-5	4

17R Respuesta D. Las reuniones de espacio abierto (Open Space Meetings) son un estilo de "desconferencia" autoorganizada donde los participantes crean y administran su propia agenda de sesiones paralelas en torno a un tema específico. Las personas proponen sesiones que les gustaría liderar, o sesiones que les gustaría ver a otros exponer. Rápidamente surge un horario, compuesto de sesiones sobre temas que las personas sienten pasión por seguir. Si un participante se encuentra en una reunión donde no puede aprender o contribuir, se trasladará a otra sala.

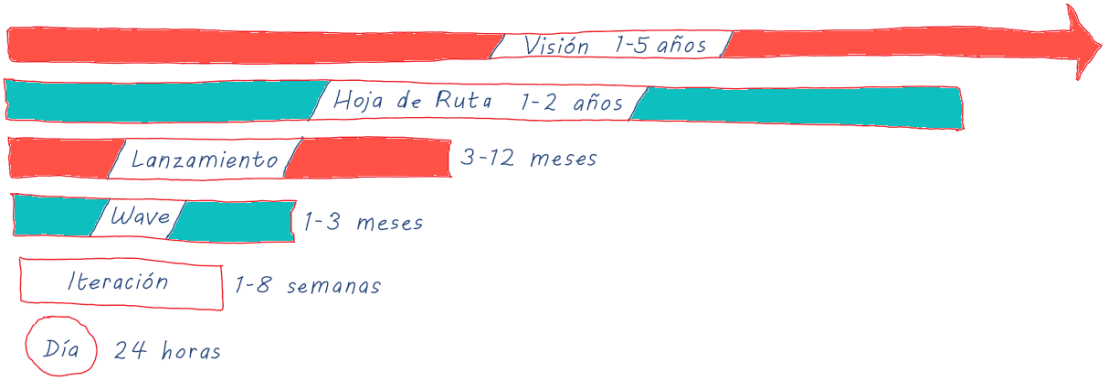
18R Respuesta B. En proyectos de metodologías ágiles es normal este tipo de situaciones de desacuerdos constructivos (divergencia) sobre diferentes puntos de vista para abordar problemas técnicos. El líder del equipo no debería intervenir de ninguna manera a menos que el conflicto comience a tornarse insano o personal.

19R Respuesta A. Gráfica de Trabajo Realizado (Burnup Chart): representación gráfica del trabajo realizado. Suele incluir una línea horizontal con el total de trabajo a realizar, que varía hacia abajo o hacia arriba cuando hay cambios en el alcance del proyecto. / Gráfica de Trabajo Pendiente (Burndown Chart): representación gráfica del trabajo pendiente frente al tiempo restante en un período de tiempo preestablecido. Suele ser confuso identificar en esa gráfica los cambios de alcance. / Gráfica de velocidad: cantidad de puntos de historia realizados por el equipo en cada iteración. No indica cambios de alcance. / Diagrama de Gantt: no suele ser una herramienta muy utilizada en proyectos ágiles.

Capítulo 5
Planificación adaptativa

5.1 Planificación ágil

La planificación de proyectos ágiles ocurre en diferentes horizontes temporales.



Nota: *Las duraciones del gráfico son solo ejemplos y varían según el tipo de proyecto*.

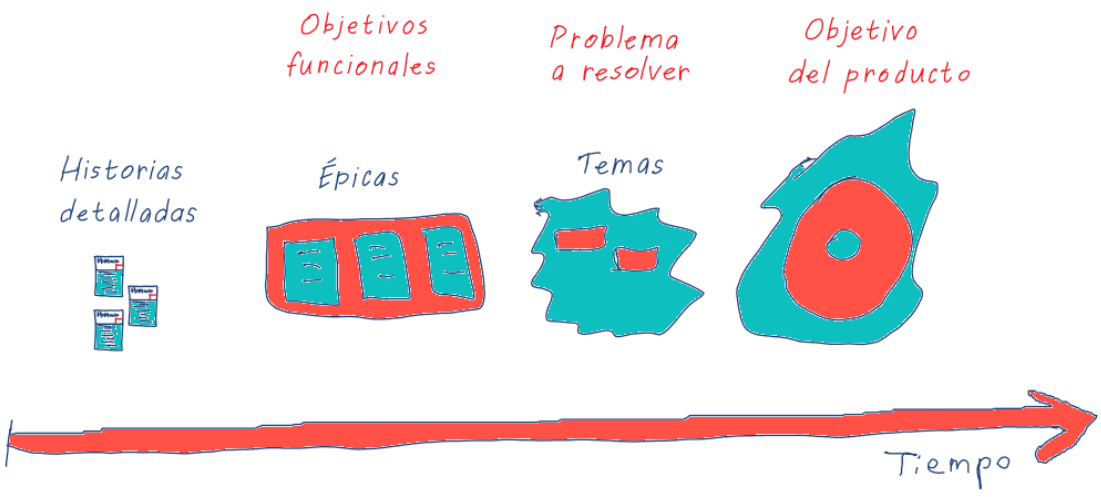
- **Visión del producto**: qué es el producto, quién lo usará y por qué lo usará. Cómo el producto respalda la estrategia de la organización. Revisión anual.
- **Hoja de ruta del producto**: mapa con los requisitos de alto nivel del producto y los plazos para los lanzamientos. Sirve tanto para alinear a los interesados como para adquirir el presupuesto del proyecto. Revisión semestral o cuando lo solicite el cliente.
- **Lanzamiento**: cronograma de alto nivel para los lanzamientos sobre las funcionalidades del producto. Revisión trimestral.
- **Wave**: épicas e historias de usuario detalladas para las próximas iteraciones y a nivel más agregado para lo que se podría realizar más adelante. Revisión mensual.
- **Iteración**: objetivos, tareas y requisitos de la iteración. Revisión al final de cada iteración.
- **Día**: actividades por realizar durante un día de trabajo. Revisión diaria.

Elaboración progresiva (Rolling wave planning)

La elaboración progresiva es un proceso iterativo que incrementa el nivel de detalle de un plan a medida que se cuenta con mayor cantidad de información y con estimaciones más precisas.

Se planifica en detalle todo aquello que esté próximo a llevarse a cabo y se replanifica más adelante aquellas actividades que podrían ocurrir a futuro. En otras palabras, los paquetes de trabajo adquieren mayor nivel de detalle cuanto más cerca estén de convertirse en actividades actuales.

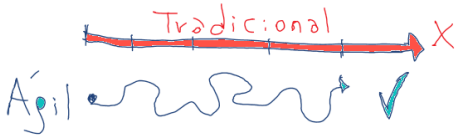
Por ejemplo, dividir las épicas en historias de usuarios y definir los detalles de esas historias a medida que avanza la planificación del lanzamiento y están más cerca de ser agregadas esas historias a una iteración.



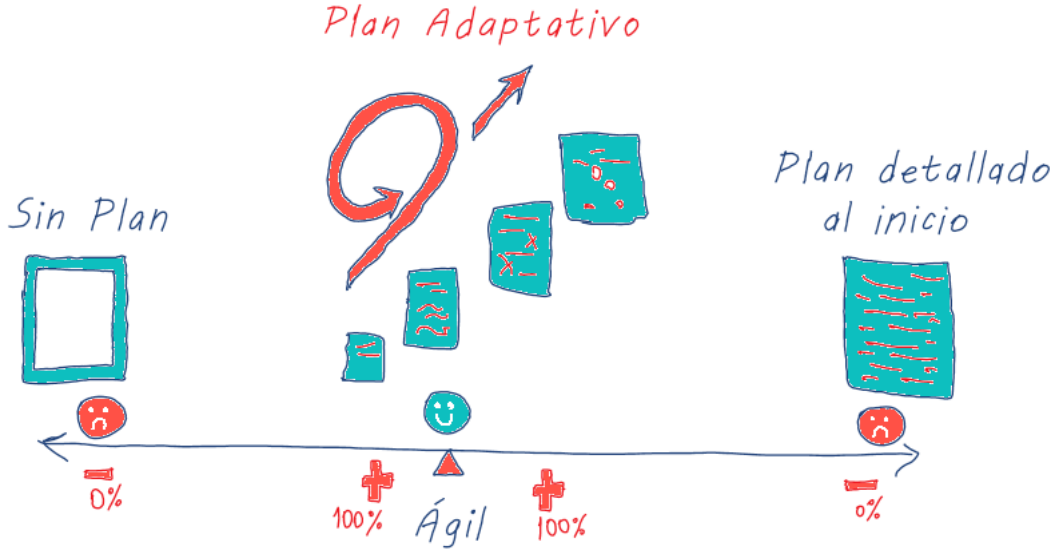
Con este tipo de planificación iterativa, se aprovechan las lecciones aprendidas de cada iteración para seguir ajustando el plan de manera continua.

5.2 Adaptabilidad

Las metodologías ágiles ponen más énfasis en la adaptabilidad de los planes con relación a la previsibilidad de seguir un plan al pie de la letra.



Los cambios de requisitos son un aspecto natural, inevitable y hasta deseable del desarrollo de proyectos ágiles. Adaptarse a los cambios de requisitos en cualquier momento del proyecto es más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos luego para controlar los cambios en los requisitos.



Mediante el desarrollo iterativo construimos una versión parcial del producto y luego expandimos esa versión a través de cortos períodos de desarrollo sucesivos seguidos de revisiones y adaptaciones. Por su parte, en el desarrollo incremental o entrega basada en funcionalidades se producen características del producto final que puedan implementarse al final de una o más de las iteraciones.

En las metodologías ágiles se suele cambiar el plan cada vez que se aprende algo nuevo o cuando es necesario mitigar un error para reflejar los cambios en los requisitos del proyecto, las prioridades, las necesidades de los interesados y los factores ambientales.

Timebox

La técnica del timebox (periodo de tiempo preestablecido) consiste en fijar el tiempo máximo de un intervalo de tiempo para conseguir unos objetivos, tomar una decisión o realizar tareas.

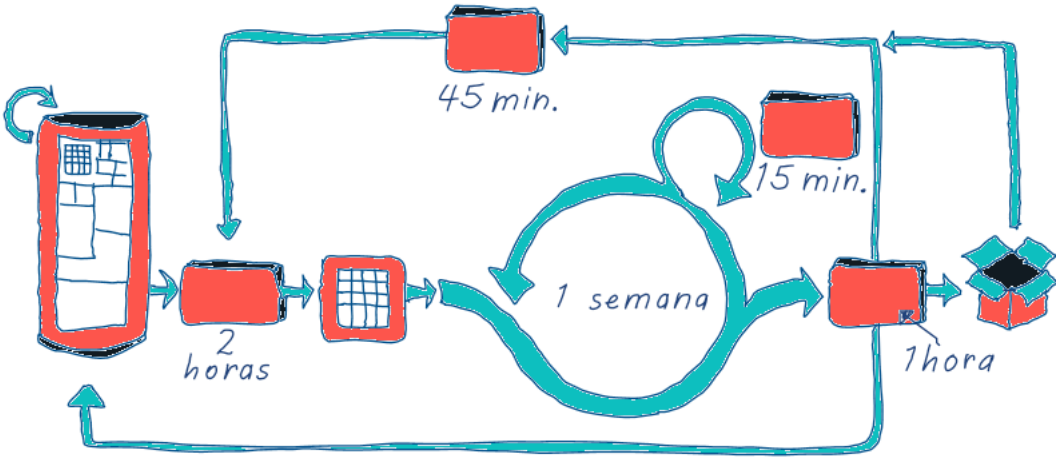
Una vez fijado este intervalo de tiempo (ej. iteración de 15 días), el equipo debe hacer lo mejor que pueda en ese plazo. De esta manera, en lugar de que el equipo trabaje en algo hasta completarlo, se acuerda de antemano que tienen un tiempo limitado para desarrollar los entregables.

La aceptación del equipo de esta restricción temporal favorece la priorización de objetivos y tareas ajustando el alcance del producto en función del plazo disponible.

Las iteraciones de tiempo preestablecido en los proyectos ágiles son fundamentales para poder implementar las adaptaciones de manera ordenada, ya que al finalizar cada iteración se utilizan las lecciones aprendidas que podrían cambiar el plan de la próxima iteración.

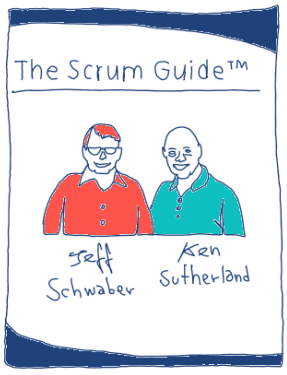
Por ejemplo, en Scrum las iteraciones tienen un plazo de entre 1 y 4 semanas y todos los eventos o reuniones tienen un periodo de tiempo preestablecido:

- Planificación de la iteración (Sprint planning) – 2 horas por semana de iteración
- Reunión diaria (Daily Scrum) – 15 minutos
- Revisión de la iteración (Sprint review) - 1 hora por semana de iteración
- Retrospectiva (Sprint retrospective) - 45 minutos por semana de iteración



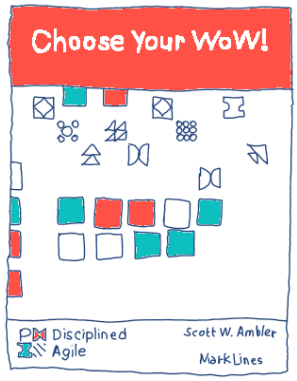
Adaptación de procesos (Tailoring)

Las metodologías ágiles no están pensadas para ser de una "talla única" que aplique a todos los proyectos, sino que los procesos ágiles pueden modificarse según las características de cada proyecto como el tamaño del equipo, contexto, recursos disponibles, criticidad, etc.

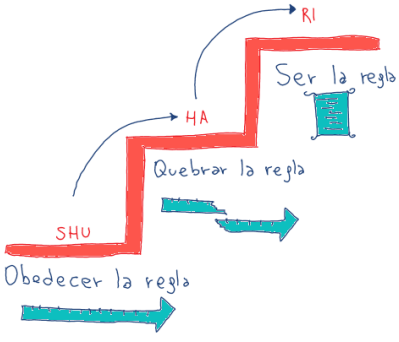


Algunos puristas ágiles piensan que el "tailoring" no es recomendable. En la Guía Scrum se menciona: "*Los roles, eventos, artefactos y reglas de Scrum son inmutables y, aunque es posible implementar solo partes de Scrum, el resultado no es Scrum. Scrum existe solo en su totalidad y funciona bien como un contenedor para otras técnicas, metodologías y prácticas*". Por ejemplo, si un equipo decide eliminar las reuniones de retrospectiva porque creen que no agregan valor, podría ocurrir que la herramienta "Scrum" dejara de funcionar correctamente.

En el otro extremo, Disciplined Agile (DA), a diferencia de otras metodologías que se presentan como un marco metodológico con pasos a seguir, ofrece opciones para adaptar los marcos existentes a cada situación en particular, brindando consejos sobre cuándo y cómo aplicar las diferente metodologías. DA fue diseñado para ofrecer un equilibrio entre los métodos con un enfoque demasiado estrecho (ej. Scrum) o aquellos demasiado detallados (ej. AUP).



En la mayoría de los proyectos ágiles los equipos deben intentar evaluar cuidadosamente la situación antes de tomar la decisión de adaptar un marco ágil. Primero deberíamos comprender las razones detrás de una práctica específica antes de decidir que no la necesitamos.



La filosofía Shu-Ha-Ri puede ser de ayuda al momento de tomar la decisión si adaptar o no una metodología.

Por último, como parte de la filosofía ágil, las mejoras o adaptación del proceso se pueden plantear en la retrospectiva de cada iteración, para implementar las mejoras en la siguiente iteración.

5.3 Temas, Épicas e Historias

El trabajo por desarrollar en el proyecto se suele dividir en temas, épicas e historias de usuario.

Mapa de historias: diagrama de temas, épicas e historias que indica las secuencias de las características opcionales que se lanzarán con el tiempo.

Temas: una colección de épicas o historias de usuario relacionadas.

Épicas: gran historia de usuario que no puede completarse en una iteración. Las épicas se dividen en historias de usuario más pequeñas para completar en distintas iteraciones.

Historias de usuario: breve descripción escrita que representa un requisito o funcionalidad del proyecto utilizando el lenguaje común del usuario.

Tareas: explican cómo se van a realizar las historias de usuario. Suelen tener una duración máxima de 16 horas.

Historias de usuario

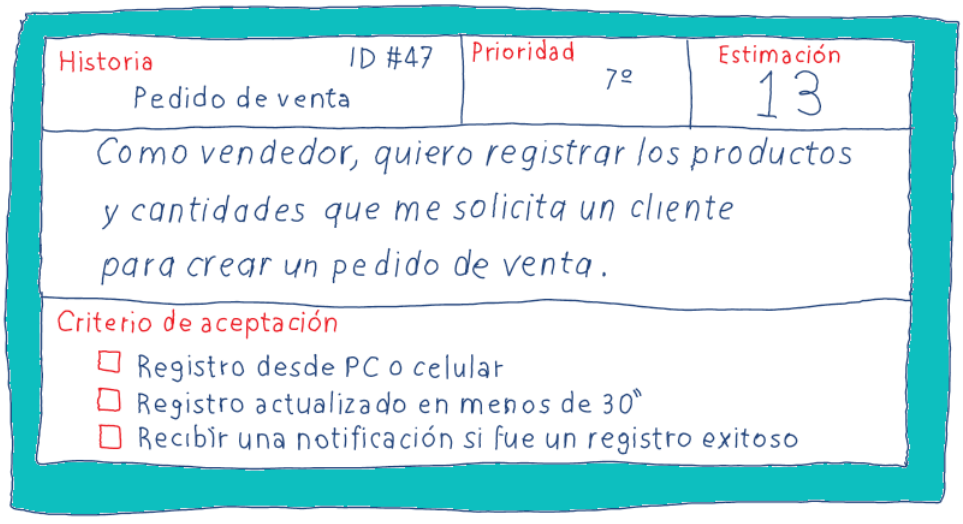
Las historias de usuarios son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente.

La historia define a alto nivel un requisito, conteniendo suficiente información para que los desarrolladores puedan estimar el esfuerzo para implementar dicho requisito.

Las historias de usuarios a menudo se escriben en tarjetas o notas adhesivas y se organizan en tableros, mesas o software de colaboración para facilitar la planificación y el debate.

Las discusiones sobre las historias de usuario son más importantes que cualquier texto escrito.

Las historias suelen seguir una plantilla simple: Como <quién> Quiero <qué> Para <objetivo>. Por ejemplo: "*Como vendedor, quiero registrar los productos y cantidades que me solicita un cliente para crear un pedido de venta*".


Historia
ID #47
Pedido de venta
Prioridad
7º
Estimación
13
Como vendedor, quiero registrar los productos y cantidades que me solicita un cliente para crear un pedido de venta.
Criterio de aceptación
Registro desde PC o celular
Registro actualizado en menos de 30"
Recibir una notificación si fue un registro exitoso

Las historias de usuario deberían enfocarse más en cómo los usuarios pueden cumplir sus objetivos, en lugar de los detalles técnicos. Este sería un mal ejemplo de una historia de usuario: "*Como cliente necesito completar un formulario de registro hecho con jQuery para registrarse online y activar la cláusula de LOPD para el envío de correos*".

Una buena historia de usuario describe solamente la tarea que nuestro usuario quiere realizar y su finalidad (para qué).

Las historias de usuario deberían tener seis atributos:

I **Independiente**: debe ser auto explicable por sí misma, no debe depender de otras historias de usuario para entender su significado.

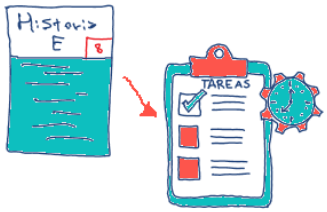
N **Negociable**: en cualquier momento antes de una iteración una historia puede ser modificada o cambiada por otra.

V **Valiosa**: deben aportar algún valor al cliente o usuario final.

E **Estimable**: se puede estimar el tamaño relativo y los recursos necesarios para completar la historia.

S **Small** (Pequeña): simples para que puedan ser priorizadas y asignadas a un equipo para que las completen en una iteración.

T **Testeable**: debe proveer claramente los criterios de verificación y aprobación, para que el equipo los aplique para evaluar si la historia está completa.



Las historias de usuario se descomponen en tareas.

Las tareas deben ser específicas, medibles, alcanzables, relevantes y de tiempo limitado.

S **Specific** (Específico): suficientemente clara para que todos puedan entenderla.

M **Medible**: ¿Hace lo que se pretende?

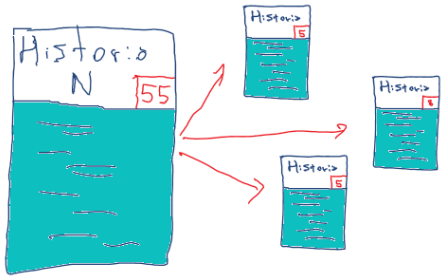
A **Alcanzable**: ¿Es razonable la meta?

R **Relevante**: Cada tarea debe contribuir a la historia en cuestión.

T **Tiempo fijo** (time-boxed): limitada a una duración específica.

Descomponer grandes historias de usuario

Cuando una historia es tan grande que no se puede completar en una iteración, deberíamos descomponerla en menores partes para crear historias de usuario más pequeñas que pueden completarse en una iteración.

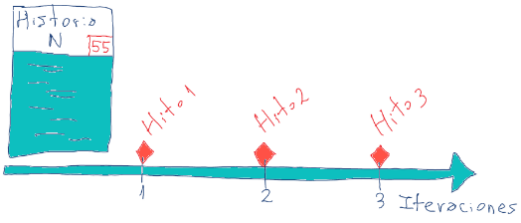


Generalmente se espera que el tiempo estimado de cada historia de usuario se sitúe entre unas 10 horas y un par de semanas. Estimaciones mayores a dos semanas son indicativo de que la historia es muy grande y debe ser dividida en historias más pequeñas. [22]

Luego de la descomposición en historias más pequeñas, el dueño del producto debe volver a decidir cuáles serán las historias para realizar durante la próxima iteración. El dueño del producto podría decidir incluir o no algunas de esas historias más pequeñas en la próxima iteración, como también podría priorizar realizar otras historias del listado de trabajo pendiente en lugar de esas nuevas historias más pequeñas.

Historias complejas

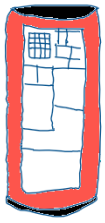
Las historias complejas no se pueden dividir en menores historias de usuario. Sin embargo, deberíamos evitar que una historia permanezca abierta durante varias iteraciones para no dejar trabajo incompleto al final de una iteración.



Como es imposible dividir la historia compleja en historias funcionales más pequeñas, deberíamos identificar puntos de progreso o hitos donde los miembros del equipo han logrado algo. Por ejemplo, si un código complejo requiere tres iteraciones, un punto de progreso podría ser cuando una primera transacción se envía a otro sistema interno y un segundo punto de progreso cuando se intercambien más transacciones para detectar errores.

[22] Mike Cohn, Agile Estimating and Planning.

Trabajo pendiente asociado al producto (Product backlog)



El product backlog es una lista de características que han sido priorizadas y contiene descripciones breves sobre el producto que se va a desarrollar. Usualmente este listado comprende diferentes tipos de elementos como: requisitos funcionales y no funcionales, bugs, trabajo técnico y adquisición de conocimientos.

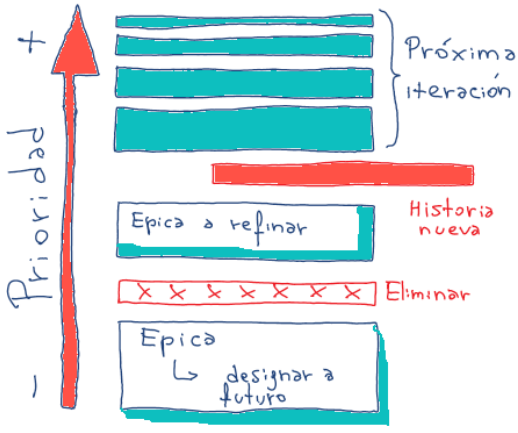
El product backlog suele estar representado por historias de usuario enumerando todas las características, funciones, requisitos, mejoras y arreglos que deben desarrollarse para el lanzamiento del producto. El product backlog debería cumplir con los atributos DEEP:

- **D**etallada: las historias deben ser comprensibles
- **E**stimable: las historias fueron estimadas por esfuerzo relativo (ej. puntos de historia)
- **E**mergente: se pueden agregar, remover o actualizar historias en cualquier momento
- **P**riorizada: las historias están priorizadas de mayor a menor importancia relativa

Refinación del trabajo pendiente asociado al producto (Refinement)

La refinación del product backlog consiste en reuniones entre el dueño del producto y el equipo de entregas para desglosar, detallar, estimar y reorganizar el product backlog.

El product backlog se sigue refinando durante todo el proyecto y siempre se pueden agregar nuevas funcionalidades y requisitos. Las historias existentes pueden modificarse, definirse con más detalle o incluso eliminarse.

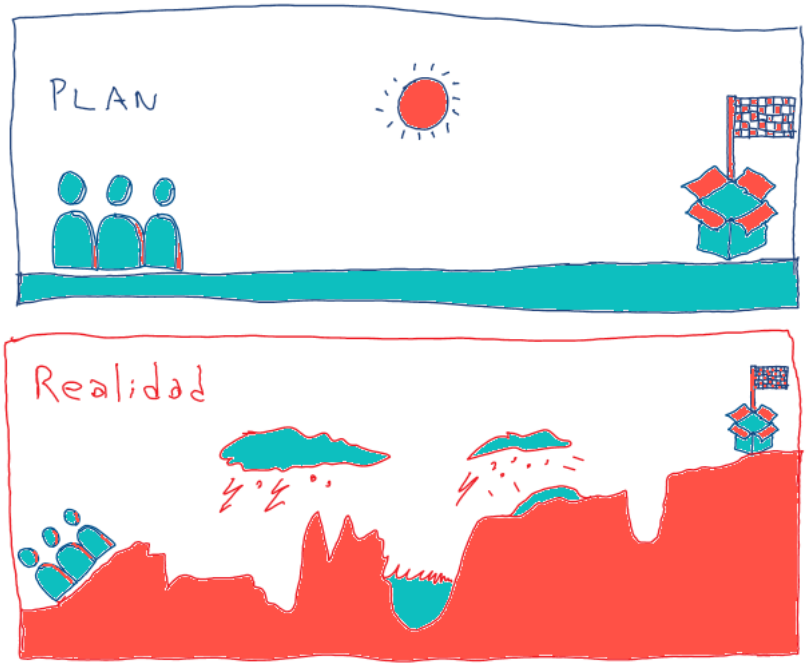


Mientras que las historias de usuario a implementar durante las próximas iteraciones deben tener buen nivel de detalle, el resto de las historias a realizar más adelante pueden estar a nivel agregado para no invertir tiempo en detallar requisitos que podría cambiar.

El refinamiento continuo del product backlog no solo se basa en el valor de cada historia de usuario, sino en el riesgo asociado a cada historia. Cuando el contexto cambia, el valor y el riesgo también cambian, por lo que las prioridades del product backlog seguramente se verán afectadas.

5.4 Técnicas de estimación

Debido a la naturaleza de los proyectos ágiles donde el alcance del proyecto se va adaptando a medida que avanza el proyecto, suele ser difícil poder estimar una fecha de finalización exacta para el lanzamiento del producto o un costo exacto del producto final.



En lugar de estimar una única fecha para un lanzamiento o para completar todas las características del proyecto y sus costos asociados, al comienzo del proyecto se pueden utilizar rangos aproximados de estimación, por ejemplo:

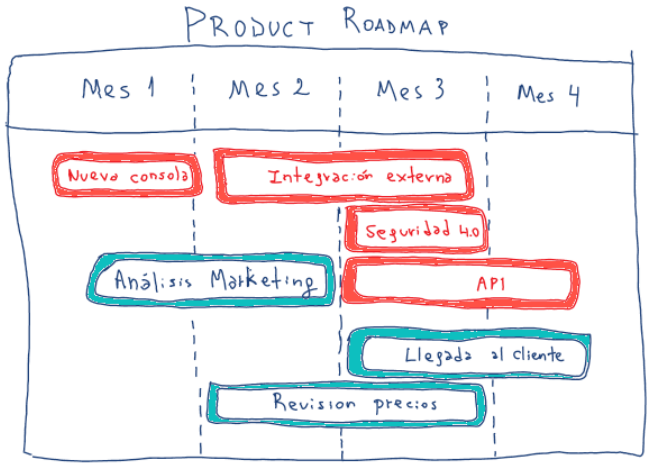
- Lanzamiento del producto en 2 meses con variación de más o menos 15 días
- Costo total del proyecto $100 más menos 50%

Mediante la planificación progresiva y adaptaciones del alcance durante el desarrollo, las estimaciones iniciales deben refinarse para proporcionar cifras más precisas basadas en la información más reciente del proyecto.

A medida que cambia el proyecto (ej. alcance, velocidad del equipo, etc.), las estimaciones deben actualizarse continuamente.

Planificación bombardeo (Blitz planning) [23]

La planificación "bombardeo" es una sesión colaborativa donde el patrocinador, el usuario principal, los analistas y los desarrolladores contribuyen juntos a desarrollar la hoja de ruta del proyecto para los próximos 3 meses.



Objetivo: enfocarse en las actividades de inicio del proyecto para conseguir una rápida autorización del patrocinador.

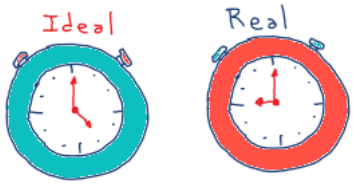
Beneficios: colaboración de los principales interesados para acordar el plan, identificando problemas técnicos por anticipado.

Pasos:

1. Reúna a los asistentes. Asegurar que el patrocinador, uno o dos usuarios clave, los analistas comerciales y todo el equipo de desarrollo estén en la sala.
2. Piense las tareas. Los participantes escriben en tarjetas las principales tareas del proyecto, con descripciones de alto nivel.
3. Ordene las tareas. Organizar las tareas según dependencias.
4. Revise las tareas. Se revisan las tareas y se agregan las que sean necesarias.
5. Estime y etiquete las tareas. Quienes vayan a realizar cada tarea escriben la estimación de plazo y los nombres de las personas necesarias para su desarrollo.
6. Lanzamiento. Identificar las primeras funcionalidades de utilidad para algunos usuarios para hacer un lanzamiento al mercado y obtener ingresos lo antes posible.
7. Hoja de ruta. Se agrupan otras tareas o funcionalidades en base a posibles lanzamientos futuros.
8. Optimice el plan. El patrocinador reconsidera la necesidad comercial del proyecto confirmando prioridades y el equipo elimina aquellas tareas que no sean necesarias.

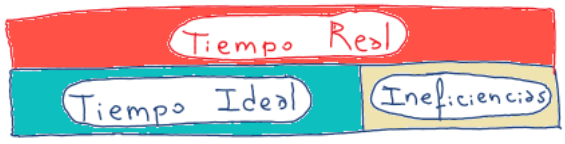
[23] Alistair Cockburn, Crystal Clear.

Tiempo ideal

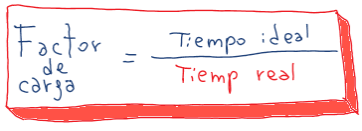


El tiempo ideal es el plazo que tomaría completar una tarea o historia de usuario suponiendo que no existirán interrupciones en los miembros del equipo.

En otras palabras, es el tiempo de trabajo en condiciones ideales suponiendo que no hay ninguna pausa por interrupciones de cuestiones ajenas a la tarea y que la persona se encuentra disponible 100% con buena concentración en la tarea.



La relación entre el tiempo ideal y el tiempo real transcurrido se lo denomina "factor de carga" (load factor).[24]



Por ejemplo, si el tiempo ideal estimado para realizar las historias de un día de trabajo es de 6 horas y el tiempo transcurrido en ese día fue de 8 horas, el factor de carga es de 0,75 (6 / 8) o 75%.

$$Tiempo\ ideal = Tiempo\ real \times factor\ de\ carga$$

Los miembros del equipo podrían enfocarse en estimar las tareas solamente en el "tiempo ideal" ya que es difícil estimar el tiempo perdido en interrupciones o imprevistos. Luego, podemos dividir esa estimación por el factor de carga para tener un estimado de la duración real que tendrá esa tarea.

$$Tiempo\ real = \frac{Tiempo\ ideal}{factor\ de\ carga}$$

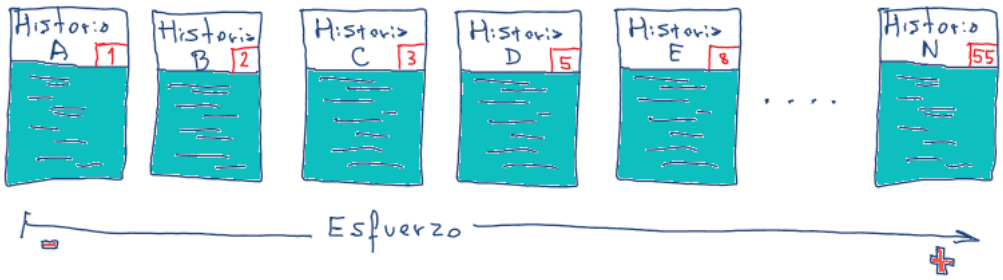
El factor de carga se puede estimar comparando datos históricos del tiempo ideal estimado para varias historias de usuario con el tiempo real en el que se completaron esas historias. Por ejemplo, las historias finalizadas la semana pasada habían estimado 20 horas ideales. Sin embargo, el tiempo transcurrido fue de 40 horas, por lo que el factor de carga fue de 0,5.

[24] Kent Beck y Martin Fowler, Planning Extreme Programming.

Puntos de historia

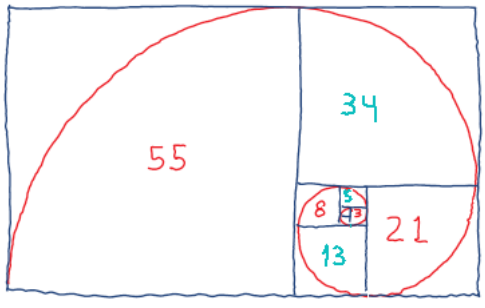
En lugar de estimar por tiempo ideal, se puede estimar en base al esfuerzo relativo necesario para completar una historia de usuario.

Los puntos de historia son un número que indica la complejidad relativa de las historias de usuario. Por ejemplo, a una historia de mínimo esfuerzo se le podrían asignar 1 punto, otra historia de esfuerzo medio 8 puntos y una historia de gran esfuerzo podría tener 55 puntos de historia.



Los puntos de historia son muy utilizados para estimar el esfuerzo relativo de las historias de usuario y esos valores no están relacionados con el tiempo ideal o el tiempo transcurrido.

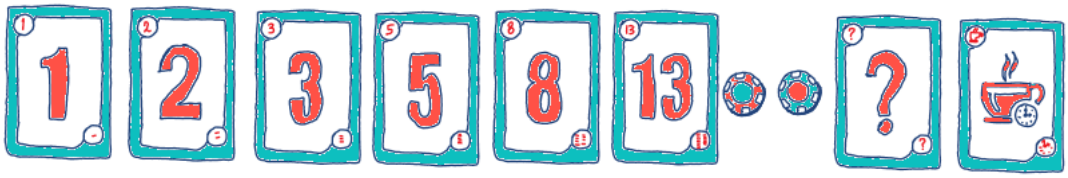
Para los valores numéricos de los puntos de historia se puede utilizar la sucesión de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 . . . , donde cada número es la sumatoria de sus dos números previos.



Póker de planificación (Planning poker)

El póker de planificación es una técnica para calcular una estimación basada en el consenso para estimar el esfuerzo o el tamaño relativo de pocas historias de usuario (ej. aprox. 15 historias) y con un número reducido de participantes (ej. máximo 8).

Los miembros del equipo utilizan una baraja de cartas numeradas de menor a mayor esfuerzo con la siguiente secuencia: 1, 2, 3, 5, 8, 13, 20, 40 y 100. Además, existen las cartas "?" (duda) y ☕ (taza de café) que significa que el participante necesita un descanso.



Para iniciar la sesión de planificación, el dueño del producto lee una historia de usuario o describe una funcionalidad. Luego, los miembros del equipo colocan sobre la mesa (o tablero electrónico si es un equipo remoto) la carta con el número de esfuerzo relativo que estiman para completar esa historia.

Todos los participantes muestran sus cartas al mismo tiempo para evitar que alguna persona influya sobre las estimaciones de otros.

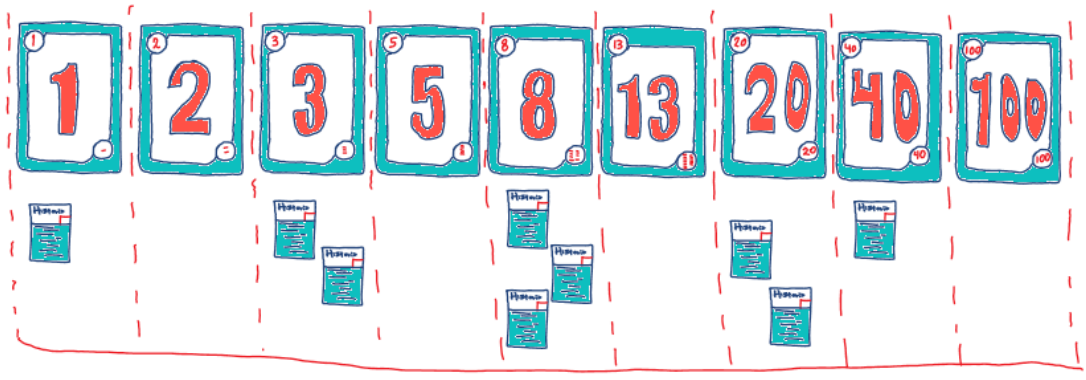
Cuando existen grandes diferencias de estimación, se produce un diálogo colaborativo para justificar las diferentes estimaciones.

La votación se repite hasta que el equipo llega a un consenso sobre la estimación.

Cuando no se llega a consenso y siguen siendo muy dispares las estimaciones de diferentes miembros del equipo, podría tener sentido no colocar esa historia entre las prioritarias a llevar a cabo ya que podría ser muy riesgosa.

Sistema de cubículos (Bucket system)

Para aquellos casos de varias historias de usuario o un gran número de participantes, se puede utilizar el sistema de cubículos que es mucho más rápido que el póker de planificación.



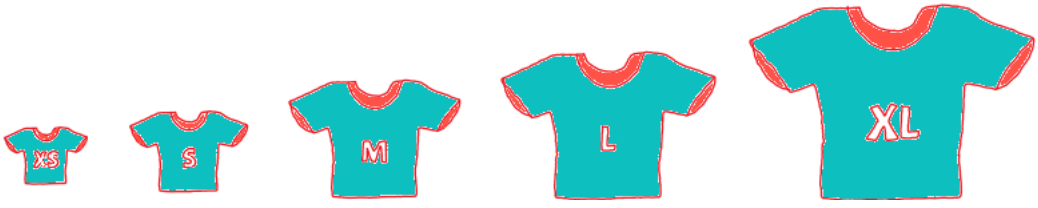
Pasos:

1. Configure el entorno físico según el diagrama de arriba donde cada carta tiene un cubículo para colocar las historias de usuario.
2. Se elije al azar una historia del product backlog y se lee al grupo. Se coloca esa historia en el cubículo "8" como primer parámetro de referencia.
3. Se elije otra historia al azar y se lee al grupo. El grupo discute y llega a un consenso de su posición relativa y la coloca en el cubículo correspondiente. Se repite lo mismo con la tercer historia seleccionada al azar.
4. Reacomode esas 3 historias según su esfuerzo relativo en los cubículos que correspondan. Por ejemplo, la primera historia que comenzó en “8”, podría reacomodarse a otro casillero.
5. Asigne todas las historias restantes por igual entre todos los participantes. Cada participante coloca sus historias en los cubículos sin discutir con otros participantes. Si una persona tiene una historia que no comprende, se la ofrece a otra persona.
6. Todos revisan en silencio las historias de los cubículos. Si alguien encuentra una historia que cree que está fuera de lugar, el grupo lo discute hasta lograr un consenso sobre su ubicación relativa.
7. Escriba los números de los cubículos en las tarjetas para registrar las estimaciones de las historias de usuario.

Talles de camisetas (T-shirt size)

Esta técnica permite estimar rápidamente el esfuerzo de un gran número de épicas o grandes historias de usuario del listado de trabajo pendiente.

Cada una de las historias o épicas se estima en función de su esfuerzo relativo utilizando un talle de camisetas: Extra Small (XS), Small (S), Medium (M), Large (L) o Extra Large (XL).



La decisión sobre el tamaño se basa en una discusión abierta y colaborativa mutua entre los miembros del equipo.

Trabajar con talles de camisetas obliga al equipo a utilizar una baja precisión en la estimación, lo que evita crear un concepto erróneo de precisión exacta en plazos y costos.

También se puede vincular los talles de camisetas con una idea aproximada del impacto en plazo, recursos y presupuesto, como en el ejemplo que se presenta en la tabla a continuación.

Talle	Plazo	Personas	Costo
XS	1-4 semanas	1-2	$1.000 - $10.000
S	2-3 meses	3-5	$10.000 - $50.000
M	4-6 meses	6-10	51.000 – 300.000
L	7-12 meses	11-20	301.000 – 1.000.000
XL	+1 año	+20	+ 1.000.000

Seguramente existen épicas que se pueden realizar en corto plazo (ej. S) pero con muchos recursos y presupuesto (ej. L), o viceversa, en esos casos se podría definir esa épica como de gran esfuerzo (L).

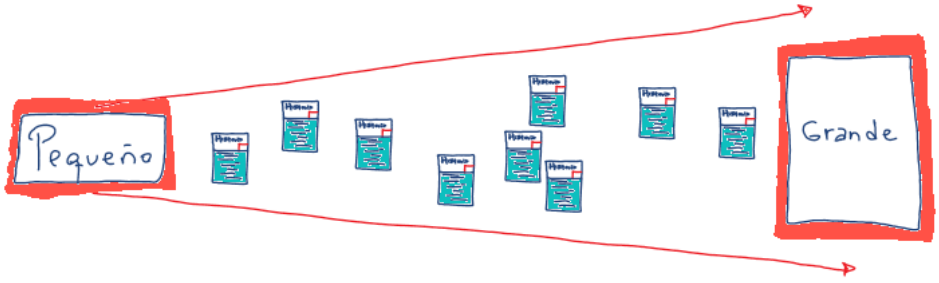
Este método es una forma rápida para tener una idea aproximada del tamaño total del listado de trabajo pendiente.

Estimación por afinidad (Affinity estimation)

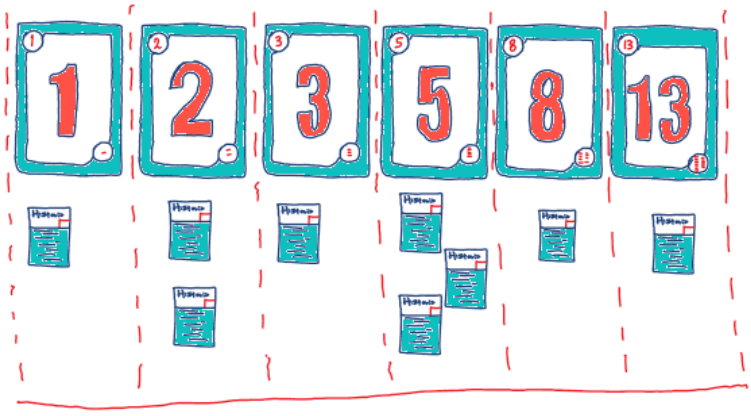
Esta técnica utiliza varios de los conceptos desarrollados previamente para estimar de manera rápida el tamaño relativo de gran cantidad de historias de usuario del product backlog.

Pasos:

1. Los miembros del equipo de desarrollo, de manera individual y en silencio, colocan las historias en una pared o tablero ordenadas según su tamaño relativo.



2. El equipo de desarrollo discute sobre los tamaños relativos del paso anterior y reorganizan las historias solicitando aclaraciones al dueño del producto.
3. Las historias se colocan en "cubículos" etiquetados. Las etiquetas podría ser el talle de camisetas (XS, S, M, L, XL) o puntos de historia (1, 2, 3, 5, 8, 13).



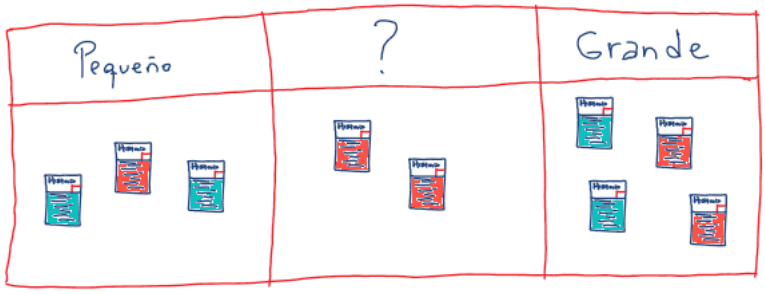
4. El dueño del producto discute los tamaños relativos estimados por el equipo. El equipo puede decidir mover alguna historia a otro cubículo.
5. Se etiquetan todas las historias de usuario con el tamaño estimado.

Pequeño / Incierto / Grande

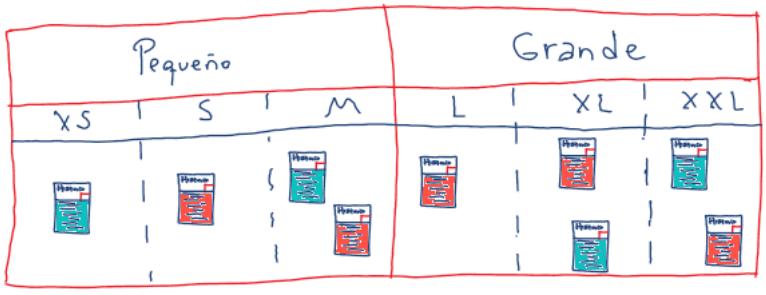
Esta es una técnica muy rápida de estimación aproximada. Se le pide al equipo que coloque las historias de usuario en una de estas categorías:

- ✓ Pequeño
- ✓ Incierto
- ✓ Grande

El primer paso es categorizar los elementos obvios en las dos categorías extremas de “grande” o “pequeño”. A continuación, el grupo puede discutir los elementos más complejos que fueron catalogados como “inciertos” para colocarlos en grande o pequeño.



Finalmente, se puede asignar tamaños relativos a estas 2 categorías (ej. Pequeño: XS, S, M; Grande: L, XL, XXL) para estimar 6 tamaños relativos de todas las historias de usuario.



Votación de puntos (Dot voting)

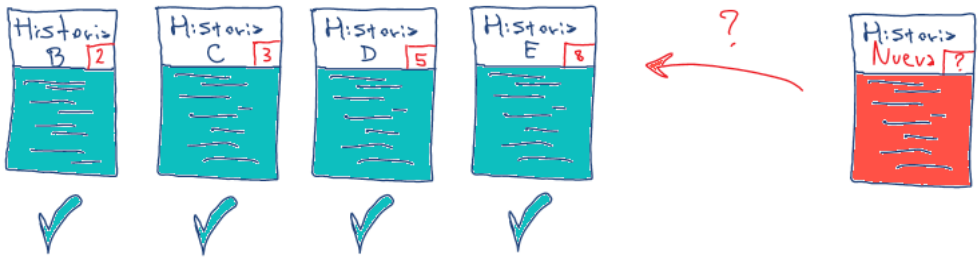
Esta técnica es simple y rápida para estimar pocas historias de usuario.

Cada persona recibe una pequeña cantidad de pegatinas que debe distribuir entre las historias de usuario en función de su esfuerzo relativo. A mayor esfuerzo, más pegatinas debería colocar en la historia.

Las historias que reciben mayor cantidad de puntos de los participantes se consideran las de mayor esfuerzo relativo.

Triangulación (Triangulating)

El equipo estima por analogía comparando la historia del usuario que desea estimar con algunas historias previamente estimadas.



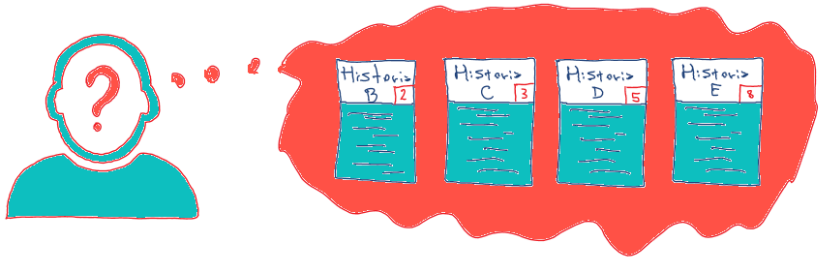
Luego deciden si la historia de usuario es aproximadamente del mismo tamaño, más pequeña o grande que las otras historias estimadas.

Esta triangulación es de gran utilidad para aplicar en cualquiera de las técnicas mencionadas previamente.

Delphi de banda ancha (Wideband Delphi Technique)

Técnica basada en el consenso para estimar el esfuerzo de una actividad.

El coach ágil o facilitador solicita de manera privada a cada uno de los miembros del equipo que hagan sus estimaciones sobre el esfuerzo relativo de las historias de usuario, manteniendo la confidencialidad de esas estimaciones.



El facilitador luego presenta las estimaciones a todo el equipo para que discutan si están de acuerdo o no con esos resultados.

Se llama "banda ancha" porque, en comparación con el método Delphi tradicional, tiene una mayor interacción y más comunicación entre los participantes que tienen la oportunidad de discutir entre todos las diferentes estimaciones.

Mapeo mudo (Mute mapping)

Luego de una tormenta de ideas entre los participantes sobre la explicación de las distintas historias de usuario, el equipo trabaja en silencio para agrupar las historias en diferentes categorías (ej. talle de camisetas).

Las estimaciones se consideran completas cuando cada participante deja de mover las historias y señala su satisfacción con las agrupaciones.

Esta herramienta de diagramación por afinidad fomenta la participación equitativa y reduce el sesgo de tener solamente opiniones de algunos miembros del equipo.

5.5 Planificar iteraciones

La planificación de iteraciones es un evento en el que todos los miembros del equipo determinan qué parte del trabajo pendiente pueden comprometerse a entregar durante la próxima iteración.

Los resultados de la planificación de la iteración serán:

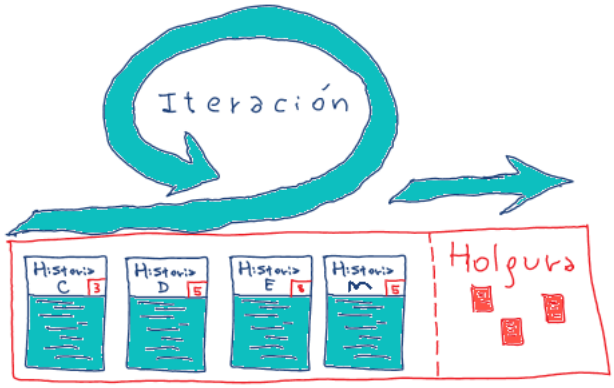
El listado de trabajo pendiente de la iteración (sprint backlog) que incluye las historias comprometidas y sus criterios de aceptación.

Una declaración de objetivos comerciales de la iteración.

Compromiso del equipo con el trabajo necesario para alcanzar los objetivos.

Holgura (Slack)

Debido a la variabilidad inherente en las estimaciones, se pueden agregar holguras en la planificación de iteraciones con tareas de baja prioridad que se podrían eliminar en cualquier momento si el equipo se atrasara en las tareas prioritarias.



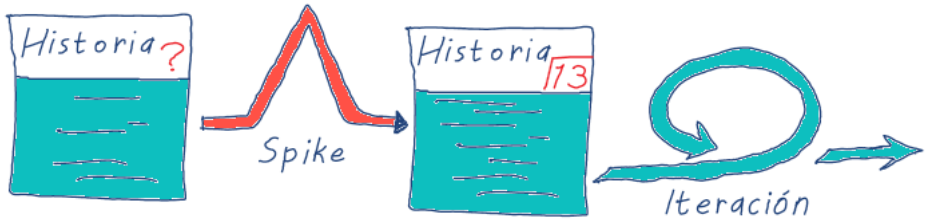
Por lo tanto, si hubiera un retraso en alguna de las tareas prioritarias, al tener estas holguras planificadas, se podría cumplir con los pronósticos de estimación.

Ejemplos de holguras:

- Agregar pruebas a un código heredado
- Pagar una deuda técnica mejorando el sistema de construcción
- Refactorización arquitectónica
- Tiempo para investigación

Spike (Ápice o pico)

Cuando nos enfrentamos con una historia de usuario a la que es difícil poder estimarla (ej. cuántos puntos de historia) por falta de información o incertidumbre, se puede realizar un spike para obtener más información y luego estimar la historia de usuario.



Un spike es un tipo de historia de usuario de muy corto plazo utilizado para investigación, diseño, exploración y creación de prototipos.

Objetivos:

- ✓ Obtener conocimiento para reducir el riesgo de un enfoque técnico
- ✓ Comprender mejor un requisito
- ✓ Aumentar la confiabilidad de una estimación de la historia

Spikes funcionales: analizar el comportamiento general de la solución y determinar cómo desglosarla, cómo organizar el trabajo, dónde existe riesgo y complejidad y cómo utilizar ideas para influir en las decisiones de implementación.

Spikes técnicos: investigar varios enfoques en la solución. Por ejemplo: decisión de construir versus comprar; evaluar el rendimiento potencial o el impacto de la carga de una nueva historia de usuario; evaluar enfoques de implementación técnica específicos; desarrollar confianza sobre la alternativa de solución deseada; etc.

Ejemplos de spike:

- Estimar nuevas características y capacidades para analizar el comportamiento implícito de una funcionalidad.
- Proporcionar información sobre el enfoque para dividir una historia en piezas más pequeñas y cuantificables.
- Realizar análisis de viabilidad de una épica.
- Realizar investigación básica para familiarizarlos con una nueva tecnología.

Los spikes no agregan valor directo al usuario, por lo que deberían utilizarse con moderación.

Sashimi

En el contexto de los métodos ágiles, cada rebanada del sashimi se entiende como un pedacito del todo que está listo para consumirse en cualquier momento.

Cuando se planean las actividades de una iteración, los productos resultantes deben estar listos para consumo del usuario. Eso quiere decir que no sólo debemos contar con el sistema de software, sino que también debe estar lista la documentación, el instalador, el manual de usuario, etc.

En otras palabras, todas las partes del entregable deben estar completas al finalizar la iteración o el lanzamiento.

5.6 Ejercicios – Planificación

1 Ordene los diferentes componentes de planificación de menor duración (1) a mayor duración (6), colocando la duración aproximada.

Componente de planificación	Orden (1 a 6)	Duración
Día		
Hoja de ruta		
Iteración		
Lanzamiento		
Visión		
Wave		

2 Conteste verdadero o falso con relación al rolling wave planning.

Frase	V	F
La elaboración progresiva es un proceso iterativo que disminuye el nivel de detalle de un plan a medida que se cuenta con mayor cantidad de información.		
La elaboración progresiva es un proceso iterativo que incrementa el nivel de detalle de un plan a medida que se cuenta con estimaciones menos precisas.		
Los paquetes de trabajo adquieren menor nivel de detalle cuanto más cerca estén de convertirse en actividades actuales.		
Se aprovechan las lecciones aprendidas de cada iteración para seguir ajustando el plan de manera continua.		
Se planifica a nivel agregado todas las actividades del proyecto.		

3 Conteste verdadero o falso con relación a la adaptabilidad del plan.

Frase	V	F
Adaptarse a los cambios de requisitos en cualquier momento del proyecto es más realista que intentar definir todos los requisitos al comienzo del proyecto.		
En el desarrollo iterativo se producen características del producto final que puedan implementarse al final de una o más de las iteraciones.		
Los cambios de requisitos son un aspecto inevitable y hasta deseable del desarrollo de proyectos ágiles.		
Mediante el desarrollo incremental construimos una versión parcial del producto y luego expandimos esa versión a través de cortos períodos de desarrollo sucesivos seguidos de revisiones y adaptaciones.		

4 Según Scrum, coloque el timebox recomendado de los siguientes eventos para un proyecto que ha definido iteraciones de 2 semanas.

Evento	Timebox
Planificación de la iteración	
Reunión diaria	
Revisión de la iteración	
Retrospectiva de la iteración	

5 Conteste verdadero o falso con relación al tailoring.

Frase	V	F
Disciplined Agile ofrece opciones para adaptar los marcos existentes a cada situación en particular.		
Primero deberíamos comprender las razones detrás de una práctica específica antes de decidir que no la necesitamos.		
Según la Guía Scrum, los roles, eventos, artefactos y reglas son adaptables y flexibles para cada organización.		
Según la Guía Scrum, Scrum se puede utilizar de manera parcial y funciona bien como un contenedor para otras técnicas, metodologías y prácticas.		

6 Complete con una palabra.

Las épicas se dividen en _____

Las historias se dividen en _____

Los temas se dividen en _____

7 Complete las palabras de los atributos INVEST de las historias de usuario.

I _____

N _____

V _____

E _____

S _____

T _____

8 Complete las palabras de los atributos SMART de las tareas.

S _____

M _____

A _____

R _____

T _____

9 Conteste verdadero o falso con relación a las historias de usuario.

Frase	V	F
Cuando una historia demore más de 8 horas hay que descomponerla en historias más pequeñas.		
En las historias complejas deberíamos identificar puntos de progreso o hitos en las distintas iteraciones.		
Las historias complejas hay que dividirlas en menores historias de usuario.		
Si una historia no se puede completar en una iteración deberíamos descomponerla en historias más pequeñas.		

10 Complete los atributos DEEP del trabajo pendiente asociado al producto.

D _____

E _____

E _____

P _____

11 Conteste verdadero o falso con relación al refinamiento del product backlog.

Frase	V	F
El product backlog se sigue refinando durante todo el proyecto y siempre se pueden agregar nuevas funcionalidades y requisitos.		
El refinamiento continuo del product backlog no solo se basa en el valor de cada historia de usuario, sino en el riesgo asociado a cada historia.		
Las historias de usuario existentes pueden modificarse, definirse con más detalle o incluso eliminarse.		
Se llevan a cabo reuniones entre el dueño del producto y el equipo de entregas para desglosar, detallar, estimar y reorganizar el product backlog.		

12 Complete del 1 al 8 los pasos para Blitz planning.

Paso	Descripción
	Estime y etiquete las tareas. Plazo recursos necesarios de cada tarea.
	Hoja de ruta. Agrupar posibles lanzamientos futuros.
	Lanzamiento. Identificar las primeras funcionalidades de utilidad.
	Optimice el plan. El patrocinador reconsidera la necesidad comercial del proyecto.
	Ordene las tareas. Organizar las tareas según dependencias.
	Piense las tareas. Los participantes escriben en tarjetas las principales tareas.
	Reúna a los asistentes.
	Revise las tareas. Se revisan las tareas y se agregan las que sean necesarias.

13 Conteste verdadero o falso con relación al tiempo ideal.

Frase	V	F
El tiempo ideal es el plazo que tomaría completar una tarea o historia de usuario suponiendo que no existirán interrupciones en los miembros del equipo.		
La relación entre el tiempo ideal y el tiempo real transcurrido se lo denomina “factor de carga” (load factor).		
Tiempo ideal = Tiempo real / factor de carga		
Tiempo real = Tiempo ideal x factor de carga		

14 Complete los 3 números que faltan de los puntos de historia.

0, 1, 2, 3, __ , 8, __ , 21, __ , 55.

15 Conteste verdadero o falso con relación al póker de planificación.

Frase	V	F
Cartas que se utilizan: 1, 2, 3, 5, 8, 13, 20, 40, 100, ? y ☕		
Cuando existen grandes diferencias de estimación, se utiliza la estimación más alta.		
Los participantes muestran sus cartas de a uno por vez hasta que participen todos.		
SI fueran muy dispares las estimaciones se puede colocar esa historia entre las prioritarias.		
Sirve para estimar muchas historias de usuario con un gran número de participantes.		

16 Complete del 1 al 7 los pasos para el bucket system.

Paso	Descripción
	Asigne todas las historias restantes por igual entre los participantes. Cada participante coloca sus historias en los cubículos sin discutir con otros.
	Configure el entorno físico con cubículos para colocar las historias de usuario.
	Escribir los números de los cubículos en las tarjetas para registrar las estimaciones de las historias de usuario.
	Reacomodar las primeras 3 historias según su esfuerzo relativo en los cubículos que correspondan.
	Se elije al azar una historia del product backlog y se lee al grupo. Se coloca esa historia en el cubículo "8" como primer parámetro de referencia.
	Se elije otra historia al azar y llega a un consenso de su posición en el cubículo correspondiente. Se repite lo mismo con la tercer historia seleccionada al azar.
	Todos revisan en silencio las historias de los cubículos. Si alguien encuentra una historia fuera de lugar, el grupo busca un consenso sobre su ubicación relativa.

17 Conteste verdadero o falso con relación al talle de camisetas.

Frase	V	F
Cada una de las historias o épicas se estima en función de su esfuerzo relativo utilizando un talle de camisetas.		
La decisión sobre el tamaño se basa en la información anónima de los miembros del equipo.		
Permite estimar rápidamente el esfuerzo de un gran número de épicas o grandes historias de usuario del listado de trabajo pendiente		
Se puede vincular los talles de camisetas con una idea aproximada del impacto en plazo, recursos y presupuesto.		
Trabajar con talles de camisetas obliga al equipo a utilizar una baja precisión en la estimación.		

18 Complete del 1 al 5 los pasos para la estimación por afinidad.

Paso	Descripción
	El dueño del producto discute los tamaños relativos estimados por el equipo. El equipo puede decidir mover alguna historia a otro cubículo.
	El equipo de desarrollo discute sobre los tamaños relativos del paso anterior y reorganizan las historias solicitando aclaraciones al dueño del producto.
	Las historias se colocan en "cubículos" etiquetados. Las etiquetas podría ser el talle de camisetas (XS, S, M, L, XL) o puntos de historia (1, 2, 3, 5, 8, 13).
	Los miembros del equipo de desarrollo, de manera individual y en silencio, colocan las historias en una pared o tablero ordenadas según su tamaño relativo.
	Se etiquetan todas las historias de usuario con el tamaño estimado.

19 Complete cuál de las siguientes técnicas de estimación corresponde a cada caso: (Pequeño / Incierto / Grande); Dot voting; Triangulación; Wideband Delphi; Mute mapping

Frase	Técnica
El equipo estima por analogía comparando la historia del usuario que desea estimar con algunas historias previamente estimadas.	
Las estimaciones se consideran completas cuando cada participante deja de mover las historias.	
Las historias que reciben mayor cantidad de puntos de los participantes se consideran las de mayor esfuerzo relativo.	
Se mantiene la confidencialidad de esas estimaciones de cada miembro del equipo.	
Se puede asignar tamaños relativos a las 2 grandes categorías.	

20 Conteste si los siguientes ítems corresponden a “slack”, “spike” o “sashimi”

Ejemplo	Planificación
Agregar pruebas a un código heredado	
Contar con el sistema de software, la documentación, el instalador y el manual de usuario al finalizar la iteración.	
Estimar nuevas características y capacidades para analizar el comportamiento implícito de una funcionalidad.	
Pagar una deuda técnica mejorando el sistema de construcción	
Proporcionar información sobre el enfoque para dividir una historia en piezas más pequeñas y cuantificables.	
Realizar análisis de viabilidad de una épica.	
Realizar investigación básica para familiarizarlos con una nueva tecnología.	
Refactorización arquitectónica	
Tiempo para investigación	

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

21 Adler ha comenzado un "Design Sprint" de cinco días facilitando técnicas de divergencia entre los miembros de su equipo ágil con el objetivo de buscar la mayor cantidad de opciones posibles. El juicio del equipo se ha aplazado durante dos días para favorecer la creatividad e innovación. Al tercer día, Adler coordina con los interesados para que lleguen a una única solución convergente del prototipo que van a testear. ¿Qué herramienta podría haber utilizado Adler para la toma de decisiones convergente?

A. Scamper
B. Comprar una característica (Buy a feature)
C. Crazy 8
D. Escritura de ideas (Brainwritting)

22 Hinato y los miembros de su equipo están refinando el listado de trabajo pendiente para comenzar con la tercera iteración de una semana en un proyecto que optimiza la carga de baterías. ¿Cuál de los siguientes ítems es correcto?

A. Las historias del listado de trabajo pendiente se estimaron con una granularidad similar
B. No se deberían agregar nuevas funcionalidades en la cuarta iteración
C. Las potenciales historias de la sexta iteración no tienen detalle
D. Después de la refinación del listado de trabajo pendiente, no deberían eliminarse historias de usuario

23 Marlene es la coach ágil de un proyecto que quiere desarrollar sensores para mejorar la calidad del aire, pero el equipo de desarrollo tiene dudas de cómo lograr ese objetivo y cuántos cambios tecnológicos podrían avecinarse. ¿Qué debería hacer Marlene?

A. Liderazgo adaptativo en base a los cambios tecnológicos
B. Planificar solamente a nivel de épicas hasta que exista menor incertidumbre
C. Elaboración progresiva sobre el detalle de las historias de usuario
D. Sashimi detallado para cada épica

5.7 Respuesta ejercicios – Planificación

Horizonte de planificación

Componente de planificación	Orden (1 a 6)	Duración
Día	6	24 horas
Hoja de ruta	2	1 año
Iteración	5	15 días
Lanzamiento	3	6 meses
Visión	1	3 años
Wave	4	3 meses

Rolling wave planning.

Frase	V	F
La elaboración progresiva es un proceso iterativo que disminuye el nivel de detalle de un plan a medida que se cuenta con mayor cantidad de información.		√
La elaboración progresiva es un proceso iterativo que incrementa el nivel de detalle de un plan a medida que se cuenta con estimaciones menos precisas.		√
Los paquetes de trabajo adquieren menor nivel de detalle cuanto más cerca estén de convertirse en actividades actuales.		√
Se aprovechan las lecciones aprendidas de cada iteración para seguir ajustando el plan de manera continua.	√	
Se planifica a nivel agregado todas las actividades del proyecto.		√

Adaptabilidad

Frase	V	F
Adaptarse a los cambios de requisitos en cualquier momento del proyecto es más realista que intentar definir todos los requisitos al comienzo del proyecto.	√	
En el desarrollo iterativo se producen características del producto final que puedan implementarse al final de una o más de las iteraciones.		√
Los cambios de requisitos son un aspecto inevitable y hasta deseable del desarrollo de proyectos ágiles.	√	
Mediante el desarrollo incremental construimos una versión parcial del producto y luego expandimos esa versión a través de cortos períodos de desarrollo sucesivos seguidos de revisiones y adaptaciones.		√

Timebox

Evento	Timebox
Planificación de la iteración	4 horas
Reunión diaria	15 minutos
Revisión de la iteración	2 horas
Retrospectiva de la iteración	1,5 horas

Tailoring

Frase	V	F
Disciplined Agile ofrece opciones para adaptar los marcos existentes a cada situación en particular.	√	
Primero deberíamos comprender las razones detrás de una práctica específica antes de decidir que no la necesitamos.	√	
Según la Guía Scrum, los roles, eventos, artefactos y reglas son adaptables y flexibles para cada organización.		√
Según la Guía Scrum, Scrum se puede utilizar de manera parcial y funciona bien como un contenedor para otras técnicas, metodologías y prácticas.		√

Temas, épicas, historias

Las épicas se dividen en historias

Las historias se dividen en tareas

Los temas se dividen en épicas

Atributos de historias de usuario.

Independiente

Negociable

Valiosa

Estimable

Small (Pequeña)

Testeable

Atributos de las tareas.

Specific (Específico)

Medible

Alcanzable

Relevante

Tiempo fijo

9R

Frase	V	F
Cuando una historia demore más de 8 horas hay que descomponerla en historias más pequeñas.		√
En las historias complejas deberíamos identificar puntos de progreso o hitos en las distintas iteraciones.	√	
Las historias complejas hay que dividirlas en menores historias de usuario.		√
Si una historia no se puede completar en una iteración deberíamos descomponerla en historias más pequeñas.	√	

10R Atributos del product backlog

Detallada

Estimable

Emergente

Priorizada

11R Refinamiento

Frase	V	F
El product backlog se sigue refinando durante todo el proyecto y siempre se pueden agregar nuevas funcionalidades y requisitos.	√	
El refinamiento continuo del product backlog no solo se basa en el valor de cada historia de usuario, sino en el riesgo asociado a cada historia.	√	
Las historias de usuario existentes pueden modificarse, definirse con más detalle o incluso eliminarse.	√	
Se llevan a cabo reuniones entre el dueño del producto y el equipo de entregas para desglosar, detallar, estimar y reorganizar el product backlog.	√	

12R Blitz planning

Paso	Descripción
5	Estime y etiquete las tareas. Plazo recursos necesarios de cada tarea.
7	Hoja de ruta. Agrupar posibles lanzamientos futuros.
6	Lanzamiento. Identificar las primeras funcionalidades de utilidad.
8	Optimice el plan. El patrocinador reconsidera la necesidad comercial del proyecto.
3	Ordene las tareas. Organizar las tareas según dependencias.
2	Piense las tareas. Los participantes escriben en tarjetas las principales tareas.
1	Reúna a los asistentes.
4	Revise las tareas. Se revisan las tareas y se agregan las que sean necesarias.

13R Tiempo ideal

Frase	V	F
El tiempo ideal es el plazo que tomaría completar una tarea o historia de usuario suponiendo que no existirán interrupciones en los miembros del equipo.	√	
La relación entre el tiempo ideal y el tiempo real transcurrido se lo denomina “factor de carga” (load factor).	√	
Tiempo ideal = Tiempo real / factor de carga		√
Tiempo real = Tiempo ideal x factor de carga		√

14R Fibonacci para puntos de historia.

0, 1, 2, 3, **5** , 8, **13** , 21, **34** , 55.

15R Póker de planificación

Frase	V	F
Cartas que se utilizan: 1, 2, 3, 5, 8, 13, 20, 40, 100, ? y ☕	√	
Cuando existen grandes diferencias de estimación, se utiliza la estimación más alta.		√
Los participantes muestran sus cartas de a uno por vez hasta que participen todos.		√
SI fueran muy dispares las estimaciones se puede colocar esa historia entre las prioritarias.		√
Sirve para estimar muchas historias de usuario con un gran número de participantes.		√

16R Bucket system

Paso	Descripción
5	Asigne todas las historias restantes por igual entre los participantes. Cada participante coloca sus historias en los cubículos sin discutir con otros.
1	Configure el entorno físico con cubículos para colocar las historias de usuario.
7	Escribir los números de los cubículos en las tarjetas para registrar las estimaciones de las historias de usuario.
4	Reacomodar las primeras 3 historias según su esfuerzo relativo en los cubículos que correspondan.
2	Se elije al azar una historia del product backlog y se lee al grupo. Se coloca esa historia en el cubículo "8" como primer parámetro de referencia.
3	Se elije otra historia al azar y llega a un consenso de su posición en el cubículo correspondiente. Se repite lo mismo con la tercer historia seleccionada al azar.
6	Todos revisan en silencio las historias de los cubículos. Si alguien encuentra una historia fuera de lugar, el grupo busca un consenso sobre su ubicación relativa.

17R Talle de camisetas

Frase	V	F
Cada una de las historias o épicas se estima en función de su esfuerzo relativo utilizando un talle de camisetas.	√	
La decisión sobre el tamaño se basa en la información anónima de los miembros del equipo.		√
Permite estimar rápidamente el esfuerzo de un gran número de épicas o grandes historias de usuario del listado de trabajo pendiente	√	
Se puede vincular los talles de camisetas con una idea aproximada del impacto en plazo, recursos y presupuesto.	√	
Trabajar con talles de camisetas obliga al equipo a utilizar una baja precisión en la estimación.	√	

18R Estimación por afinidad.

Paso	Descripción
4	El dueño del producto discute los tamaños relativos estimados por el equipo. El equipo puede decidir mover alguna historia a otro cubículo.
2	El equipo de desarrollo discute sobre los tamaños relativos del paso anterior y reorganizan las historias solicitando aclaraciones al dueño del producto.
3	Las historias se colocan en "cubículos" etiquetados. Las etiquetas podría ser el talle de camisetas (XS, S, M, L, XL) o puntos de historia (1, 2, 3, 5, 8, 13).
1	Los miembros del equipo de desarrollo, de manera individual y en silencio, colocan las historias en una pared o tablero ordenadas según su tamaño relativo.
5	Se etiquetan todas las historias de usuario con el tamaño estimado.

19R Técnicas de estimación

Frase	Técnica
El equipo estima por analogía comparando la historia del usuario que desea estimar con algunas historias previamente estimadas.	Triangulación
Las estimaciones se consideran completas cuando cada participante deja de mover las historias.	Mute mapping
Las historias que reciben mayor cantidad de puntos de los participantes se consideran las de mayor esfuerzo relativo.	Dot voting
Se mantiene la confidencialidad de esas estimaciones de cada miembro del equipo.	Wideband Delphi
Se puede asignar tamaños relativos a las 2 grandes categorías.	Pequeño / Incierto / Grande

20R "Slack", "Spike", "Sashimi"

Ejemplo	Planificación
Agregar pruebas a un código heredado	Slack
Contar con el sistema de software, la documentación, el instalador y el manual de usuario al finalizar la iteración.	Sashimi
Estimar nuevas características y capacidades para analizar el comportamiento implícito de una funcionalidad.	Spike
Pagar una deuda técnica mejorando el sistema de construcción	Slack
Proporcionar información sobre el enfoque para dividir una historia en piezas más pequeñas y cuantificables.	Spike
Realizar análisis de viabilidad de una épica.	Spike
Realizar investigación básica para familiarizarlos con una nueva tecnología.	Spike
Refactorización arquitectónica	Slack
Tiempo para investigación	Slack

Respuesta B. Buy a feature (comprar una característica): juego donde se crea una lista de características potenciales colocando un precio a cada una basado en los costos de desarrollo o el valor para el cliente. Los clientes compran las características que desean en la próxima versión de su producto utilizando el dinero ficticio de juego que se les dio. Los clientes utilizan todo su dinero comprando las funciones prioritarias. Como resultado de este juego los interesados llegan a una decisión convergente o consenso sobre las principales características que debería tener el producto. / Scamper, Crazy 8 y Brainwritting son técnicas para fomentar el pensamiento divergente entre los miembros del equipo. / Scamper: una vez identificado el problema, se hacen preguntas relacionadas con: Sustituir, Combinar, Adaptar, Modificar, Poner otros usos, Eliminar y Reordenar. / Crazy 8: Los miembros del equipo deben dibujar ocho ideas distintas en pocos minutos para generar una amplia variedad de soluciones a un problema. / Brainwritting (Escritura de ideas): dinámica de grupo similar al brainstorming (tormenta de ideas) donde los participantes escriben en un papel sus ideas antes de ponerlas en común con los demás. Esta técnica busca aumentar la participación de todos los interesados, en especial de los más tímidos que podrían no participar en un brainstorming.

Respuesta C. El listado de trabajo pendiente (Product Backlog) se sigue refinando durante todo el proyecto y siempre se pueden agregar nuevas funcionalidades y requisitos. Las historias existentes pueden modificarse, definirse con más detalle o incluso eliminarse. Las funcionalidades y requisitos no están completamente definidos desde el principio, sino que el product backlog se desarrolla de forma iterativa. Los ítems del product backlog tienen una granularidad diferente; mientras que los elementos a implementar durante las próximas iteraciones deben estar definidos con mayor detalle (grano fino), el resto pueden estar definidos de manera más general (grano grueso). El motivo de la granularidad diferente es que no tiene sentido invertir tiempo en la descripción detallada de los requisitos de una historia hasta que comience su implementación, ya que los requisitos suelen cambiar a medida que avanza el proyecto.

Respuesta C. Elaboración Progresiva: proceso iterativo que incrementa el nivel de detalle de un plan a medida que se cuenta con mayor cantidad de información y con estimaciones más precisas. Por ejemplo, dividir las épicas en historias de usuarios y definir los detalles de esas historias a medida que avanza la planificación del lanzamiento y están más cerca de ser agregadas a un sprint. / En el contexto de los métodos ágiles, cada rebanada del sashimi se entiende como un pedacito del todo que está listo para consumirse en cualquier momento. Cuando se planean las actividades de una iteración, los productos resultantes deben estar listos para consumo del usuario. Eso quiere decir que no sólo debemos contar con el sistema de software, sino que también debe estar lista la documentación, el instalador, el manual de usuario, etc. En otras palabras, todas las partes del entregable deben estar completas.

Capítulo 6

Resolución de problemas

6.1 Riesgos

El riesgo es un evento incierto que si ocurriese tendría un impacto negativo o positivo sobre el proyecto. Los riesgos negativos suelen ser amenazas que deberíamos mitigar y los riesgos positivos son oportunidades que podríamos aprovechar.

En este capítulo vamos a enfocarnos solamente en los riegos negativos.

En los proyectos ágiles la gestión de riesgos se lleva a cabo en cada iteración involucrando al dueño del producto, el equipo de desarrollo y los interesados clave. Al final de cada iteración se analizan las lecciones aprendidas para actualizar el plan de gestión de riesgos de la próxima iteración.

Resolver problemas una vez que ocurren es reactivo y no ayuda con la agilidad del proyecto. Deberíamos ser proactivos para anticiparnos a los problemas. Por ejemplo, identificar problemas en las reuniones de retrospectiva es una buena práctica proactiva para mitigar riesgos potenciales antes que ocurran en las próximas iteraciones.

Tipos de riesgos

Michele Sliger [25] menciona cinco riesgos principales de los proyectos ágiles:

- Variación de la productividad: diferencia entre el rendimiento planificado y el real
- Corrupción del alcance: exceso de requisitos sobre el acuerdo inicial
- Descompostura de la especificación: falta de consenso sobre los requisitos
- Defecto del cronograma: estimaciones deficientes de la duración de tareas
- Pérdida de personal: recursos humanos que se van del proyecto

Los riesgos pueden clasificarse en diferentes categorías:

Negocios: relacionados con el valor comercial.

Técnico: uso de tecnología o habilidades.

Logística: cronograma, financiación, dotación de personal, etc.

Políticos, económicos, sociales, tecnológicos, legales, ecológicos.

[25] Michele Sliger y Stacia Broderick, The Software Project Manager's Bridge to Agility.

Gestión de riesgos

Los pasos básicos para gestionar los riesgos son:

1 Identificar los riesgos mediante: lluvia de ideas, técnica Delphi, revisión de supuestos, análisis FODA (fortalezas, oportunidades, debilidades, amenazas), etc.

2 Analizar los riesgos para priorizar su importancia relativa

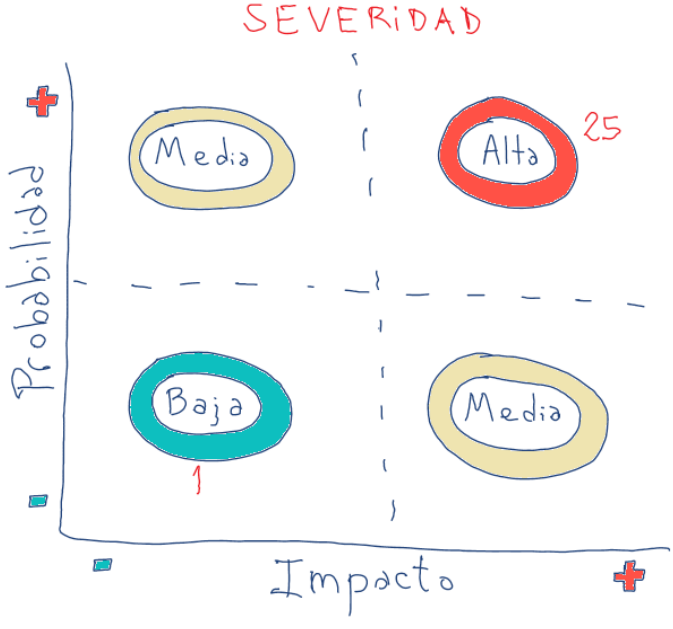
3 Planificar la respuesta a los riesgos más importantes

4 Monitorear y controlar los riesgos

Una vez identificados los riesgos, se suele realizar un **análisis cualitativo** para estimar su severidad, estimando la probabilidad de ocurrencia y el impacto.

Al multiplicar la probabilidad por el impacto se obtiene un puntaje que servirá para ordenar los riesgos desde los menos críticos hasta los más severos. Por ejemplo, en un análisis cualitativo, si se define una probabilidad de 1 (muy baja probabilidad) a 5 (muy alta probabilidad), e impacto de 1 (muy bajo impacto) a 5 (muy alto impacto), el riesgo menos grave tendrá un puntaje de 1 (1 x 1) y los más severos tendrán un puntaje de 25 (5 x 5) .

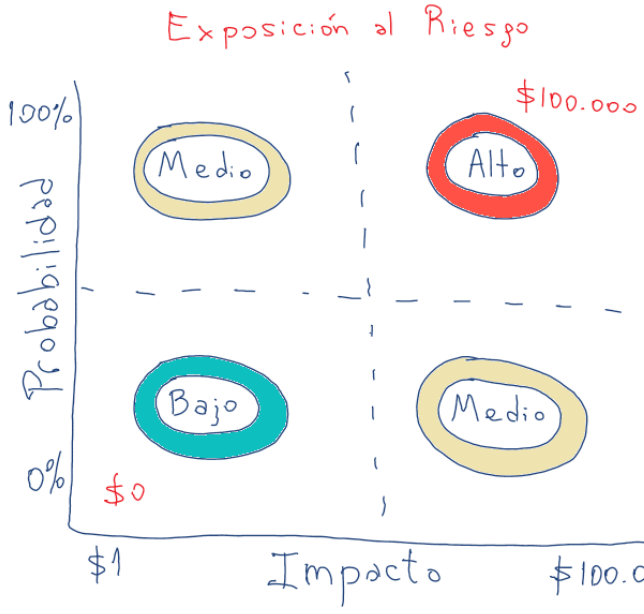
Severidad del Riesgo = Probabilidad de ocurrencia x Impacto



Por su parte, en un **análisis cuantitativo** del riesgo, la probabilidad de ocurrencia suele ser un porcentaje entre 0% y 100%, mientras que el impacto suele ser un valor monetario que representa la pérdida si el riesgo ocurre o el impacto en el cronograma del proyecto medido en días de retraso.

Por ejemplo, si existe una probabilidad del 20% de que uno de los miembros claves del equipo deje de trabajar, el impacto podría ser un retraso de una semana en el desarrollo del producto y este retraso tendría un costo monetario de $10.000.

En este caso la pérdida monetaria esperada o exposición al riesgo sería de $2.000 (20% x $10.000).



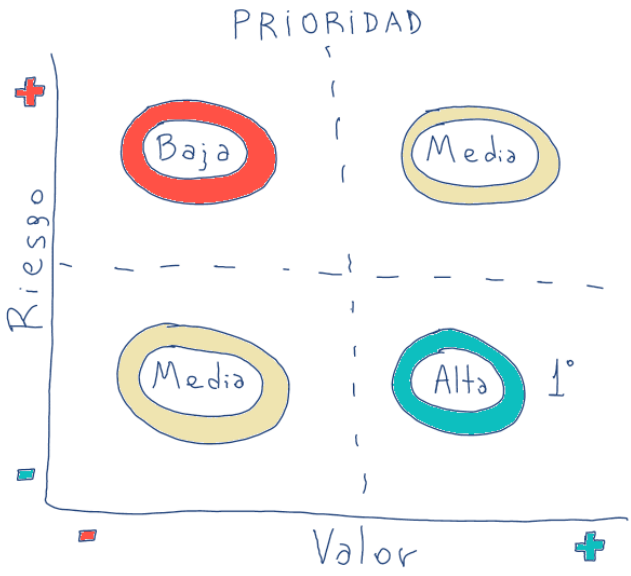
Exposición al riesgo = Probabilidad (%) x Impacto ($)

Un **censo de riesgo** es un marco simple para analizar la exposición al riesgo de un proyecto. Para cada riesgo identificado, se calcula cuantitativamente la probabilidad y el impacto (tiempo y/o costo), luego se multiplica la probabilidad por el impacto para calcular la exposición de cada riesgo. Por ejemplo, si la probabilidad de ocurrencia es 10% y el impacto un retraso de 20 días, la exposición de ese riesgo será de 2 días (10% x 20 días).

Sumando la exposición de cada riesgo identificado, se obtiene la exposición total del riesgo del proyecto.

Priorización de historias según riesgos

Las historias de usuario del product backlog deberían priorizarse no sólo en función del valor comercial, sino también del riesgo. Mientras mayor valor comercial y menor riesgo tenga una historia de usuario, mayor debería ser su prioridad.



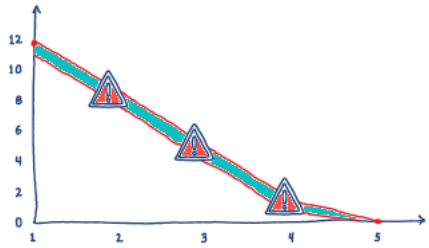
Por ejemplo, la disponibilidad de recursos suele ser un riesgo. Por lo que suele ser preferible completar una historia de menor valor comercial que tiene recursos disponibles, en lugar de dejar incompleta una historia de gran valor comercial por falta de recursos.

La situación ideal sería no realizar tareas que dependan solamente de una persona. Debemos intentar siempre tener equipos con perfil "T" o "Peine roto" para no tener que cancelar tareas si algún miembro del equipo deja de trabajar. O sea, hay que incluir en la priorización la disponibilidad de recursos o el riesgo de tener personas con perfil "I".

Las historias de usuarios de gran valor y alto riesgo deberían realizarse lo antes posible. De esa forma, si el riesgo ocurriese, tendremos holgura para gestionar la situación. Por otro lado, si esas historias se prueban en las primeras iteraciones y fallan, seguramente el costo de esa falla será menor que con el proyecto más avanzado.

Fracasar rápido permite al equipo aprender y ajustar el rumbo.

Diagrama de riesgos pendientes (Risk burndown chart)



Si bien pueden aparecer riesgos a lo largo de todo el proyecto, por lo general, a medida que avanzan las iteraciones se van completado historias del proyecto y los riesgos tienden a disminuir.

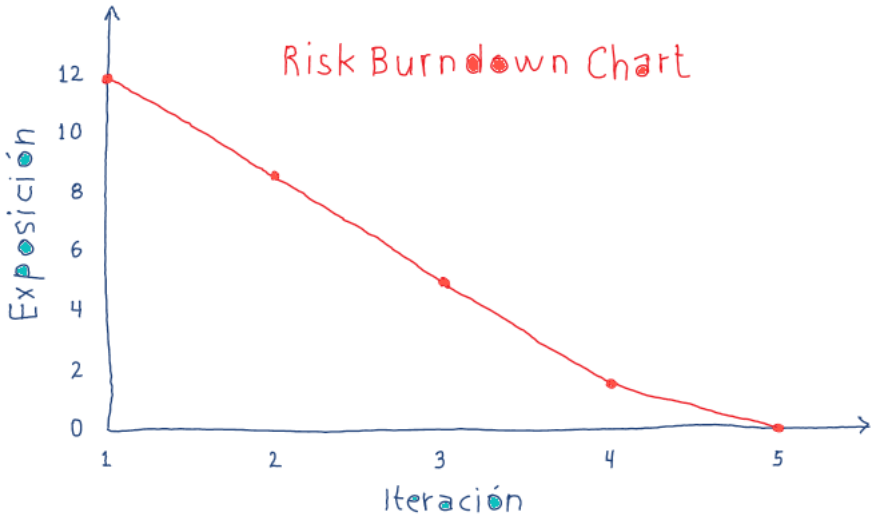
Veamos en la tabla a continuación un ejemplo de un proyecto donde se identificaron tres riesgos (a, b, c) al inicio del proyecto y todos fueron disminuyendo a medida que avanzaba el proyecto. Para estimar la exposición total del riesgo, se multiplica la probabilidad de ocurrencia por el impacto de cada riesgo (ej. días de retraso).

Iteración	Riesgo	Probabilidad	Impacto (Días de retraso)	Exposición	Exposición total
1	a	10%	20	2	12
	b	20%	30	6	
	c	40%	10	4	
2	a	8%	20	1,6	8,6
	b	15%	30	4,5	
	c	25%	10	2,5	
3	a	0%	0	0	5
	b	10%	30	3	
	c	20%	10	2	
4	a	0%	0	0	1,5
	b	0%	0	0	
	c	15%	10	1,5	
5	a	0%	0	0	0
	b	0%	0	0	
	c	0%	0	0	

Por ejemplo, en la iteración 3, la exposición total de los 3 riesgos del proyecto (a + b +c) fue: 0% x 0 días + 10% x 30 días + 20% x 10 días = 5 días.

Probabilidad de ocurrencia de 0% significa que ese riesgo ya no existe porque estaba asociado a una historia de usuario que está completa al 100%.

En el diagrama de riesgos pendientes (risk burndown chart) se grafica la exposición total del riesgo en las ordenadas y el número de iteración en la abscisa.



El diagrama de riesgos pendientes se actualiza al finalizar cada iteración para reflejar el cambio en la exposición al riesgo del proyecto. Este gráfico por lo general incluye solamente los riesgos principales (ej. top 10).

Spike para estimar riesgos

Los spikes son un tipo de historia de usuario utilizada para obtener conocimiento para reducir el riesgo de un enfoque técnico, comprender mejor un requisito o aumentar la confiabilidad de una estimación de la historia.

En un corto periodo de tiempo entre iteraciones, los desarrolladores experimentan mediante el spike para aprender sobre variables desconocidas de una historia de usuario. Por ejemplo, probar un concepto técnico. Con ese aprendizaje, se actualizan los riesgos de la historia de usuario en el product backlog para re-priorizar las historias de las próximas iteraciones.

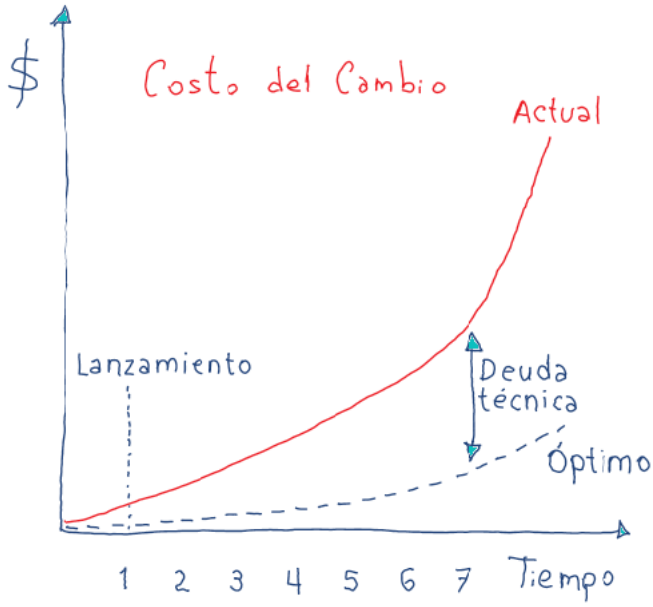
6.2 Detectar problemas

El coach ágil debe facilitar que el equipo detecte, evalúe y resuelva los problemas por sí mismo. O sea, el coach no debe resolver los problemas del equipo.

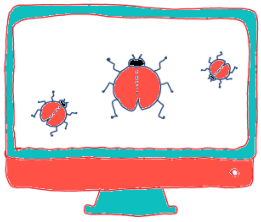
Mientras antes pueda detectar el equipo los problemas, menos costosa será su resolución. Si un problema no se detecta de manera temprana, es probable que incremente su tamaño con el correr del tiempo, lo que luego va a ocasionar retrabajos y retrasos en el cronograma del proyecto.

Deuda técnica (deuda de diseño o deuda de código)

La deuda técnica refleja el costo implícito del retrabajo adicional que haremos a futuro por elegir una solución limitada ahora en lugar de utilizar un mejor enfoque que tomaría más tiempo.



Defectos escapados (escaped defects)

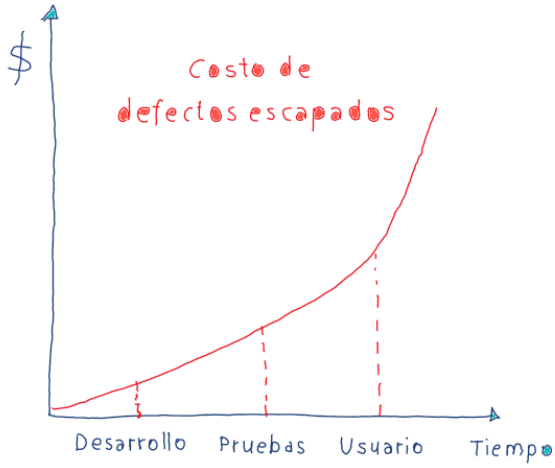


Un defecto escapado es aquel que no fue descubierto por los equipos de prueba o el dueño del producto durante la iteración, sino que fue encontrado más tarde por el usuario final durante la fase de operación.

Los defectos escapados son los más costosos de corregir, no sólo por el trabajo necesario para corregirlos o los costos incurridos en litigios, sino porque afectan de manera negativa la imagen del producto y la reputación de la organización.

En general, se gasta menos dinero cuando el equipo de desarrollo detecta el problema antes que los equipos de prueba. Por su parte, se gasta menos cuando el equipo de prueba detecta un problema en lugar del usuario final.

Por lo tanto, es beneficioso analizar las causas de los escapes para mitigar fugas futuras e impulsar la detección de defectos lo antes posible durante el proceso de desarrollo.



Los defectos escapados se monitorean a través del tiempo y son un indicador del nivel de calidad del proyecto.

El ratio error-retroalimentación es el número de nuevos defectos que tendrá el proyecto para corregir defectos existentes. Por ejemplo, 2 nuevos defectos generados para corregir 10 defectos, sería un ratio error-retroalimentación del 20% (2/10).

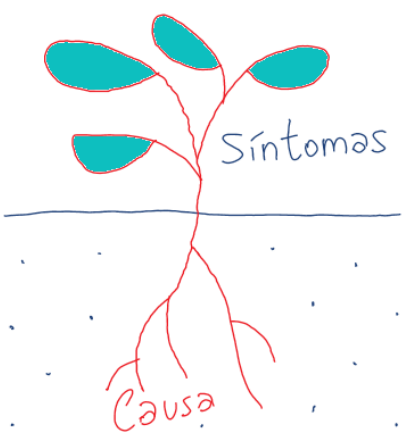
Ratio error-retroalimentación = Nuevos defectos / Defectos corregidos

5 Por qué

Los 5 ¿Por qué? es una técnica para explorar las relaciones de causa y efecto subyacentes a un problema particular para detectar la causa-raíz de ese problema.

La técnica consiste en repetir la pregunta "¿Por qué?" 5 veces, donde cada respuesta forma la base de la siguiente pregunta.

El "5" se deriva de la observación empírica en el número de iteraciones típicamente requeridas para detectar la causa raíz del problema, aunque podrían ser más o menos de 5 pregunta-respuesta como sea necesario.



Ejemplo:

1 ¿Por qué se rompió el control?
Porque la placa está defectuosa

2 ¿Por qué está defectuosa la placa?
Por mala refrigeración

3 ¿Por qué hay mala refrigeración?
Porque no llega aire

4 ¿Por qué no llega aire?
Por falta de presión

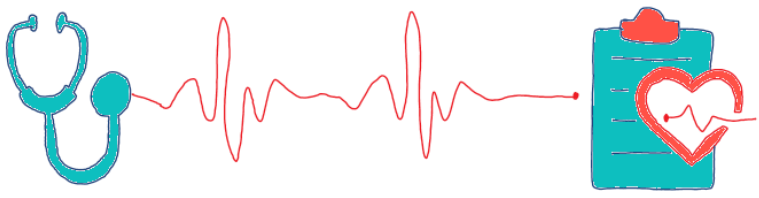
5 ¿Por qué falta presión?
Porque el filtro de aire está sucio.

Por lo tanto, la causa raíz de este problema es el filtro sucio.

Identificar la causa raíz de los problemas nos ayudará a comprender realmente cuál es el problema antes de intentar una solución rápida que podría ser sólo un parche temporal que agregará deuda técnica.

Chequeos de salud (Health checks)

Otra herramienta para detectar problemas es realizando chequeos sobre la salud del equipo.



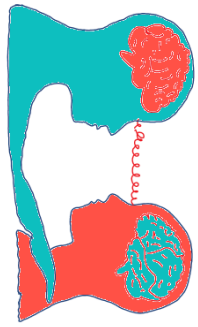
El chequeo mide el desempeño del equipo revisando su estado en temas tales como trabajo en equipo, entrega de valor, velocidad, facilidad del lanzamiento, aprendizaje, diversión, etc.

Chequeo	Equipo A	Equipo B	Equipo C
Trabajo en equipo	↓	↑	.
Entrega de valor	↑	.	↑
Velocidad	.	↑	.
Facilidad del lanzamiento	.	↓	.
Aprendizaje	↑	.	↑
Diversión	.	↑	.

Los chequeos de salud son de utilidad para enfocar los esfuerzos de mejora, detectar problemas sistémicos y ayudar a los equipos a ser más conscientes de sí mismos para que puedan desarrollarse favorablemente.

Intimidad cognitiva (Cognitive Intimacy) [26]

La colaboración real entre los miembros del equipo requiere cierta intimidad intelectual o social. La intimidad se alcanza cuando las partes manifiestan su voluntad de trabajar en una zona de confianza, vulnerabilidad, apertura, acomodación mutua y respeto.



Los equipos íntimos tienen el potencial de tener intimidad cognitiva para sentir y anticipar cómo otros miembros perciben la nueva información. Con la intimidad cognitiva se puede sentir y anticipar malentendidos a nivel individual y grupal para aclararlos rápidamente mitigando potenciales problemas de comunicación.

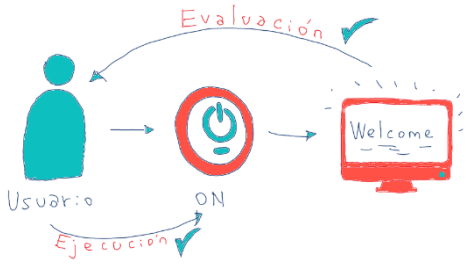
[26] Daniel Mezick, Collaborative intimacy: From good to great collaboration.

Abismo de la ejecución y la evaluación (Gulf of execution and evaluation) [27]

Podemos detectar problemas analizando los abismos de ejecución y evaluación.

Abismo de la ejecución: distancia entre lo que quiere hacer el usuario y los medios para lograrlo al interactuar con el sistema. El abismo se produce cuando el usuario sabe qué objetivos quiere lograr, pero desconoce qué pasos o acciones realizar para llevarlos a cabo.

Por ejemplo, un abismo pequeño de ejecución sería si un usuario quiere encender una computadora y sabe que apretando una tecla la computadora se enciende.



Por otro lado, un abismo grande sería cuando un usuario quiere descargar un producto online, pero tiene dudas sobre los pasos previos (alta como cliente, verificar domicilio, demostrar identidad, alta de tarjeta de crédito, verificar email, etc.).

Abismo de la evaluación: distancia entre la respuesta del sistema y la interpretación del usuario. El abismo se produce cuando el usuario es incapaz de interpretar si la acción que realizó fue exitosa o no.

Por ejemplo, un abismo pequeño sería cuando el usuario aprieta la tecla "on" y se enciende la computadora, conociendo de inmediato que su acción fue exitosa. Por otro lado, un alto abismo podría ser cuando el usuario que está intentando descargar el producto online, ve en pantalla una rueda giratoria que no indica con claridad el estado de avance de la descarga o si cometió algún error en el proceso, por lo que el usuario se queda con dudas si está realizando el proceso de descarga correctamente.

[27] Donald Norman y Stephen Draper, User-centered system design.

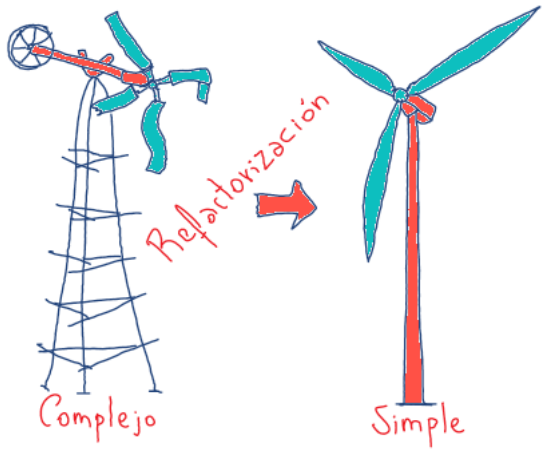
6.3 Resolver problemas

En los proyectos que utilizan metodologías ágiles, los miembros del equipo están autoorganizados y son los que están mejor preparados para encontrar soluciones a los problemas que aparezcan durante el desarrollo.

En la cultura organizacional ágil se crea un ambiente dónde se aceptan los fracasos para la mejora continua, por lo que algunos problemas podrían tener una solución desfavorable.

Cada vez que aparezca un problema, lo primero que debería hacer el coach ágil es reunir a todo el equipo (desarrolladores y dueño del producto) para compartir el problema de manera abierta y analizar alternativas de solución.

Refactorización (Refactoring)



La refactorización consiste en mejorar el diseño de un producto mejorando su disposición para el mantenimiento, sin alterar la funcionalidad esperada. En otras palabras, mejorar la operación interna de un componente sin cambiar su comportamiento externo.

En ingeniería de software la refactorización consiste en limpiar el código simplificando el código fuente sin cambiar su comportamiento. Este mantenimiento del código no arregla errores ni añade funcionalidad, sino que elimina código innecesario para facilitar el mantenimiento en el futuro.

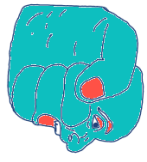
Con la refactorización continua (limpieza, mantenimiento y estandarización) dentro de cada iteración, se mantiene baja la deuda técnica, por lo que se puede extender la vida útil de la inversión.

Niveles de conflicto

Speed Leas describe cinco niveles de conflicto entre las personas [28]. A continuación, presentamos la técnica de resolución del problema para cada nivel de conflicto.

	Nivel de conflicto	Resolución
	1. Problema a resolver: orientado a problemas en lugar de orientado a la persona.	Centrarse en la colaboración y el consenso.
	2. Desacuerdo: mezcla de personalidades y problemas, el problema no puede definirse claramente.	Reunir y apoyar al equipo para una discusión.
	3. Contienda o combate: ataques personales entre bandos con dinámicas "ganar-perder".	Asegurarse de que las rivalidades personales pasen a un segundo plano y todos negocien una discusión abierta de dar y recibir.
	4. Lucha-vuelo: cambia de ganar a deshacerse de la persona enemiga, guerra de "principios" en lugar de "problemas".	Separar a los dos equipos con diplomacia e intentar resolver las diferencias antes de dejarlas para una mesa de discusión.
	5. Situaciones intratables: las personalidades se han convertido en el foco donde el conflicto es inmanejable y la energía se centra en la destrucción de los oponentes.	Intentar limitar el daño y reparar los daños colaterales.

Técnicas de resolución de problemas del equipo



- Haga varias preguntas al equipo para ayudarlos a pensar de manera diferente y buscar alternativas.
- No dedique tiempo en los pequeños problemas fáciles de resolver (síntomas), concéntrese en buscar la causa raíz del gran problema.
- Repita o parafrasee lo que escuchó para confirmar la comprensión.
- Evite inyectar sus propias ideas intentando resolver el problema para que el equipo no detenga el análisis de alternativas.

[28] Speed B. Leas, Discover Your Conflict Management Style.

CD3 (cost of delay divided by duration)

Supongamos que nos enfrentamos con el problema de tener que priorizar historias de usuario de tal forma de maximizar el valor del proyecto. Los recursos disponibles permiten llevar a cabo una sola historia por vez y la información de esas historias se presenta a continuación.

Historia	Duración (días)	Valor diario una vez completada
A	4	$1.000
B	2	$2.000
C	8	$5.000
TOTAL	**14**	**$8.000**

Cada historia completada agrega valor a la organización. Por lo tanto, cada día que pase sin completar esas historias la organización tiene un costo de oportunidad. Por ejemplo, si nos demoramos 4 días para completar la historia A, en ese plazo vamos a dejar de ganar $4.000 ($1.000 x 4 días).

¿Cuál sería el mejor criterio de priorización para minimizar el costo de oportunidad?

a) Sin priorización

Historia	Duración	Valor diario una vez completada	Demora	Costo por Demora
A	4	$ 1.000	4	$ 4.000
B	2	$ 2.000	6	$ 12.000
C	8	$ 5.000	14	$ 70.000
TOTAL	**14**			**$ 86.000**

Si decidiéramos hacer 1º A, 2º B y 3º C tendríamos un costo de oportunidad total de $86.000 calculado de la siguiente forma:

- 4 días de demora en producir A, donde dejamos de ganar $1.000 por día ($4.000 en total)
- más 6 días de demora en finalizar B (4 días de A + 2 días de B) con un costo de oportunidad de $2.000 por día ($12.000 en total)
- más 14 días de demora en completar C (4 días de A + 2 días de B + 8 días de C) dejando de ganar $5.000 por día ($70.000 en total).

b) Priorizar según duración

Con la misma lógica que el caso anterior, podemos calcular un costo de oportunidad total de $80.000 si priorizamos en función de la menor duración de cada historia (1º B, 2º A y 3º C).

Historia	Duración	Valor diario	Demora	Costo
B	2	$ 2.000	2	$ 4.000
A	4	$ 1.000	6	$ 6.000
C	8	$ 5.000	14	$ 70.000
TOTAL	**14**	**$ 8.000**		**$ 80.000**

c) Priorizar según mayor valor

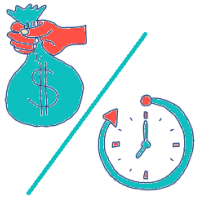
Si decidimos realizar una priorización en función del mayor valor diario de cada historia (1º C, 2º B y 3º A), el costo total de oportunidad sería de $74.000.

Historia	Duración	Valor diario	Demora	Costo
C	8	$ 5.000	8	$ 40.000
B	2	$ 2.000	10	$ 20.000
A	4	$ 1.000	14	$ 14.000
TOTAL	**14**			**$ 74.000**

d) Priorizar según CD3 (Cost of delay divided by duration).

La mejor forma de priorización sería priorizando aquellas historias de mayor CD3.

CD3 = Valor / Duración



Historia	Duración	Valor diario	CD3	Demora	Costo
B	2	$ 2.000	$ 1.000	2	$ 4.000
C	8	$ 5.000	$ 625	10	$ 50.000
A	4	$ 1.000	$ 250	14	$ 14.000
TOTAL	**14**				**$ 68.000**

Priorizando por CD3 (1º B, 2º C y 3º A), obtenemos el menor costo de oportunidad de $68.000. En otras palabras, priorizar historias según CD3 maximiza el valor del proyecto.

Depuración de escopeta (Shotgun Debugging)

La depuración de escopeta consiste en probar varias posibles soluciones de problemas al mismo tiempo, con la esperanza de que una de las soluciones funcione (ej. modificaciones del código fuente).

Esta técnica tiene una tasa de éxito relativamente baja y puede llevar mucho tiempo, excepto cuando se usa como un intento de evitar las características del lenguaje de programación que uno puede estar usando de manera incorrecta.

Otras técnicas para resolución de problemas

- Analogía: utilizar soluciones que fueron exitosas para otros problemas similares.
- Prueba y error: probar diferentes alternativas hasta que alguna resuelva el problema.
- Descomposición: partir el problema complejo en varios problemas más simples y fáciles de resolver.
- Sandboxing: crear un ambiente controlado para probar soluciones al problema, antes de intentar resolverlo con el sistema real funcionando.
- Simulación: construir un modelo para simular como podrían impactar al problema distintos escenarios.

6.4 Seguimiento y control de problemas

El seguimiento de problemas se lleva a cabo a lo largo del ciclo de vida del proyecto con el objetivo de detectar problemas lo antes posible.

El control de problemas consiste en resolver los problemas de manera proactiva a medida que se van detectando.

Las reuniones diarias y de retrospectiva suelen ser excelentes eventos para detectar problemas de manera proactiva para poder encarar su resolución de manera rápida y efectiva.

Por su parte, el análisis de la variación y tendencias, suelen ser excelentes señales de alarma para la detección temprana sobre potenciales problemas.

Indicadores rezagados (lagging) vs líderes (leading)

Los indicadores **rezagados** se basan en hechos del pasado. Suelen ser fáciles de medir, pero difíciles de mejorar.

Por otro lado, lo indicadores **líderes** son los que explican lo que podría pasar con los indicadores rezagados. Estos indicadores son más difíciles de medir, pero fáciles de mejorar.

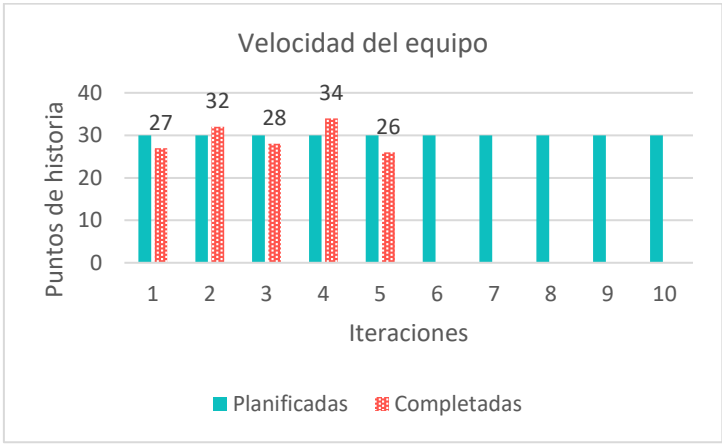
Indicador rezagado - Objetivo	Indicadores líderes
Accidentes laborales	% de trabajadores que utilizan casco % de trabajadores que respetan las normas
Rentabilidad del proyecto	Valor comercial de cada historia Velocidad del equipo
Defectos escapados	Horas de refactorización Horas de revisiones de calidad
Tiempo de ciclo	WIP Rendimiento del equipo

Para lograr los objetivos de indicadores rezagados, deberíamos prestar especial atención en los indicadores líderes. Por ejemplo, si el objetivo fuera reducir los accidentes laborales, se debería trabajar en aumentar el % de trabajadores que utilizan casco y respetan las normas.

Análisis de la variación

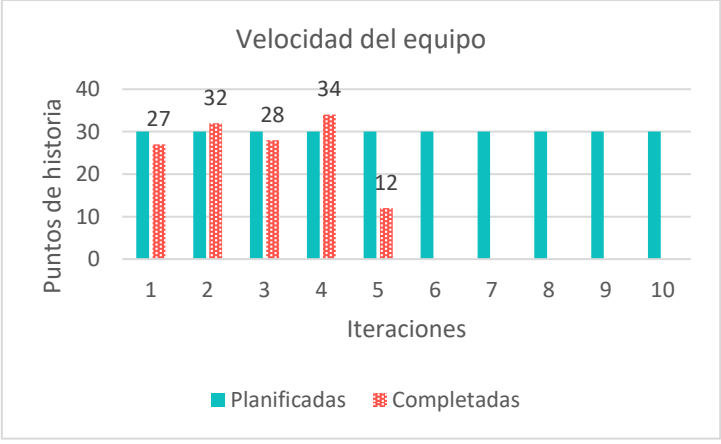
La varianza es una medida de dispersión que representa la variabilidad de una serie de datos respecto a su media. Cuando la varianza es baja, significa que los datos tienden a estar cerca de la media o valor esperado.

- **Variación común**: problema sistemático de pequeñas desviaciones. Suelen tener un impacto bajo en los resultados, por lo que no requieren acciones de control, sino que deben seguir en observación mediante un análisis de tendencias.



Por ejemplo, si la velocidad del equipo tiene una variación como la que se observa en el gráfico, podemos decir que estas pequeñas variaciones suelen ser comunes en proyectos ágiles, por lo que no se requieren acciones de control.

- **Variación especial**: desviaciones significativas que ocurren una sola vez debido a causas especiales o nuevas. Deben gestionarse de manera proactiva con acciones de control oportunas para evitar que vuelvan a ocurrir y que el proceso salga de control.



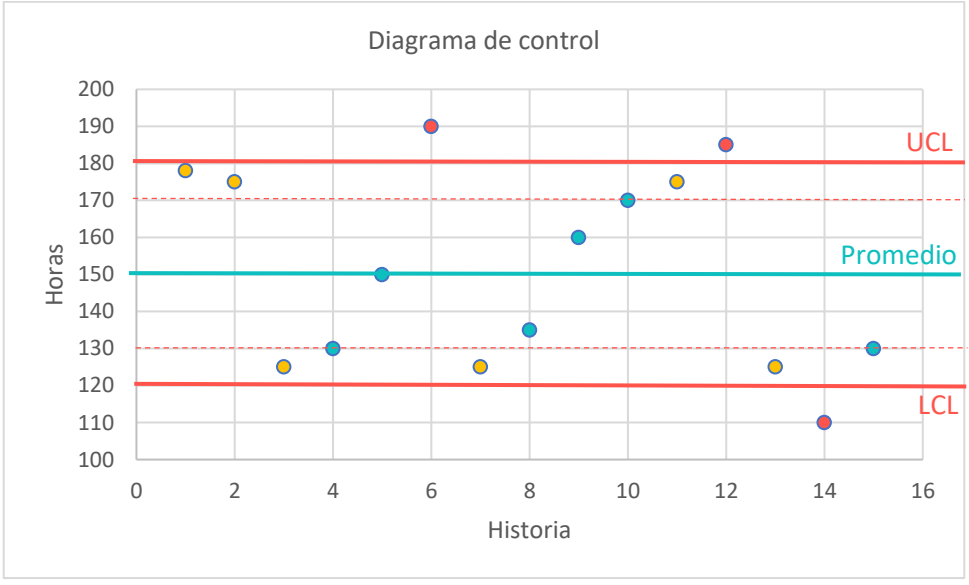
Por ejemplo, si la velocidad del equipo en la iteración 5 cae drásticamente de 34 a 12, hay que investigar la causa raíz de esa variación para implementar acciones correctivas en tiempo y forma.

Diagramas de control

Una herramienta para analizar las variaciones y decidir si se trata de una variación común o especial son los diagramas de control.

Por ejemplo, supongamos que estamos trabajando con historias de usuario de una complejidad similar donde hemos estimado que cada una debería completarse con una duración promedio de 150 horas.

El equipo podría definir para este proyecto que cualquier variación dentro de +/- 30 horas (20%) de las 150 horas estimadas sería normal. Por lo tanto, el proceso estaría dentro de variaciones normales si las historias se llevan a cabo entre las 120 y 180 horas.



UCL: upper control limit = límite superior de control ; LCL: lower control limit = límite inferior de control

Por otro lado, el equipo puede implementar alarmas tempranas que le indiquen con anticipación antes que el proceso caiga en la zona “fuera de control”. Por lo tanto, podría establecer límites de tolerancia entre las 130 y 170 horas para que fuera de ese rango se dispare alguna alerta.

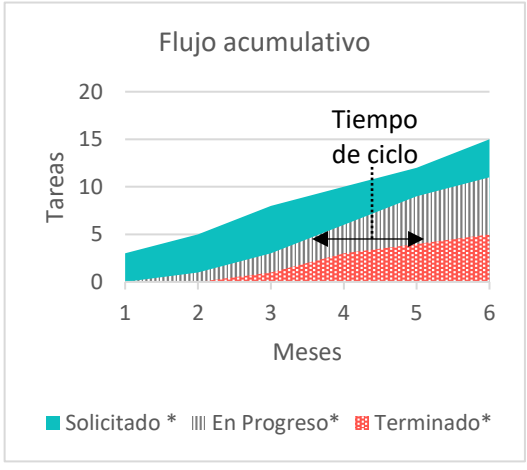
Si una historia supera el UCL, debería ser analizada inmediatamente para investigar la causa raíz del problema e implementar acciones correctivas. Por otro lado, si la historia estuviera por debajo del LCL, se debería investigar si hubo alguna buena prácticas que se pueda imitar en futuras historias.

Análisis de Tendencias

El análisis de tendencias proporciona información sobre el futuro, lo cuál es más importante para la detección de problemas que un simple análisis de la variación.

Si se detecta una tendencia negativa para el proyecto, el equipo puede predecir con anticipación problemas potenciales para adoptar acciones preventivas antes que ocurran.

Por ejemplo, en base a los datos del diagrama de flujo acumulativo, podemos proyectar la tendencia del tiempo de ciclo si no se toman acciones correctivas.



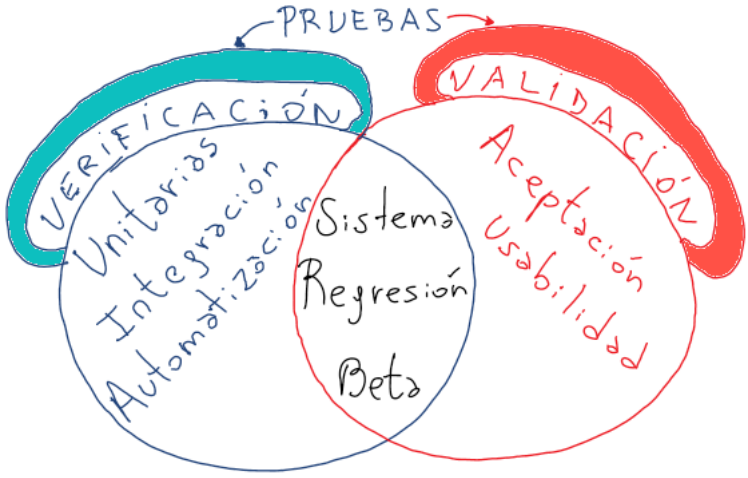
El tiempo de ciclo indica el tiempo calendario transcurrido desde que el trabajo comienza en una historia hasta que está lista para la entrega. Si aumenta el tiempo de ciclo, el número de solicitudes que entran en el tablero Kanban puede ser mayor que el número de tareas que lo estén dejando, por lo que aumenta el trabajo en progreso (WIP: Work in Progress). Por lo tanto, habrá que reconsiderar los límites del WIP existentes y concentrarse más en finalizar las tareas que están en curso antes de comenzar otras nuevas.

Tiempo de ciclo = WIP / Rendimiento

Validación y Verificación

Con el objetivo de reducir los problemas relacionados con los costos del cambio y de falla, es importante llevar a cabo de manera temprana y frecuente validaciones y verificaciones del producto.

- **Validación**: el dueño del producto evalúa si el producto cumple con las necesidades del cliente. ¿Estamos desarrollando el producto correcto?
- **Verificación**: los miembros del equipo evalúan si el producto cumple con los requisitos. ¿Estamos desarrollando el producto correctamente?



Algunas prácticas para asegurar la verificación y validación:

- Los probadores forman parte del equipo de desarrollo desde el inicio
- Los desarrolladores y probadores participan en la recopilación de requisitos del cliente
- El equipo obtiene una retroalimentación continua. Por ejemplo: reuniones diarias, revisión de la iteración, reunión de retrospectiva, lanzamiento, etc.

6.5 Ejercicios – Problemas

1 Según Michele Sliger, ¿Cuáles son los cinco riesgos principales de los proyectos ágiles?

Riesgo	
Corrupción del alcance	
Defecto del cronograma	
Descompostura de la especificación	
Financiación	
Negocios	
Pérdida de personal	
Políticos	
Uso de tecnología	
Variación de la productividad	

2 ¿Qué significan las siglas PESTLE con relación a los riesgos?

Tipo de riesgo	
P	
E	
S	
T	
L	
E	

3 Ordene de 1º a 4º los pasos para la gestión de riesgos.

Pasos	
	Analizar
	Identificar
	Monitorear
	Planificar respuesta

4 ¿Qué severidad (alta, media, baja) tendrá cada uno de los siguientes riesgos?

Riesgo	Probabilidad	Impacto	Severidad
A	Baja	Bajo	
B	Baja	Alto	
C	Alta	Bajo	
D	Alta	Alto	

5 ¿Cuál es la exposición total de riesgo del siguiente proyecto?

Riesgo	Probabilidad	Impacto	Exposición
A	10%	$5.000	
B	20%	$80.000	
C	70%	$3.000	
D	80%	$60.000	
		Total	

6 ¿Qué prioridad (alta, media, baja) tendrían las siguientes historias de usuario?

Historia	Riesgo	Valor	Prioridad
Alfa	Alto	Bajo	
Beta	Alto	Alto	
Gama	Bajo	Bajo	
Delta	Bajo	Alto	

7 Según los datos de la tabla a continuación, ¿Cuál será la exposición de riesgo del proyecto al finalizar la iteración 2?

Iteración	Riesgo	Probabilidad	Impacto (Días de retraso)	Exposición	Exposición total
1	a	10%	20		
	b	20%	30		
	c	40%	10		
2	a	8%	20		
	b	20%	20		
	c	30%	5		

8 Conteste verdadero o falso con relación a la detección de problemas.

Frase	V	F
Con la intimidad cognitiva se puede sentir y anticipar malentendidos a nivel individual y grupal.		
El ratio error-retroalimentación es el número de nuevos defectos que tendrá el proyecto para corregir defectos existentes.		
La deuda técnica refleja el costo del retrabajo a realizar hoy para mitigar futuros problemas.		
Los 7 ¿Por qué? es una técnica para explorar las relaciones de causa y efecto subyacentes a un problema particular para detectar la causa-raíz de ese problema.		
Los defectos escapados son los más costosos de corregir.		
Los defectos escapados son un indicador del nivel de calidad del proyecto.		
Se gasta más dinero cuando el equipo de desarrollo detecta el problema antes que el usuario final.		
Un defecto escapado es aquel que fue descubierto por el equipos de prueba.		
Una herramienta para detectar problemas es realizando chequeos sobre la salud del equipo.		

9 Señale si los siguientes ítems corresponden al abismo de la ejecución o de la evaluación.

Abismo	Ejecución	Evaluación
Distancia entre la respuesta del sistema y la interpretación del usuario.		
Distancia entre lo que quiere hacer el usuario y los medios para lograrlo al interactuar con el sistema.		
El usuario es incapaz de interpretar si la acción que realizó fue exitosa o no.		
El usuario sabe qué objetivos quiere lograr, pero desconoce qué pasos o acciones realizar para llevarlos a cabo.		
El usuario tiene dudas si está realizando el proceso correcto para utilizar un producto.		
Un usuario tiene dudas sobre los pasos a realizar para interactuar con un producto.		

10 ¿Cuáles de los siguientes ítems están relacionados con la refactorización?

Ítem	
Cada vez que aparezca un problema, el coach ágil debe reunir a todo el equipo para compartir el problema y analizar alternativas de solución.	
Eliminar el código innecesario para facilitar el mantenimiento en el futuro.	
En la cultura organizacional ágil se crea un ambiente dónde se aceptan los fracasos para la mejora continua.	
Limpiar el código simplificando el código fuente sin cambiar su comportamiento.	
Limpieza, mantenimiento y estandarización del código en cada iteración, para mantener baja la deuda técnica.	
Los miembros del equipo son los que están mejor preparados para encontrar soluciones a los problemas que aparezcan durante el desarrollo.	
Mejorar el diseño de un producto mejorando su disposición para el mantenimiento, sin alterar la funcionalidad esperada.	
Mejorar la operación interna de un componente sin cambiar su comportamiento externo.	

11 Unir con flechas la resolución de conflicto recomendada para cada nivel de conflicto.

Nivel de conflicto	Resolución
Contienda o combate	Asegurarse de que las rivalidades personales pasen a un segundo plano y todos negocien una discusión abierta.
Desacuerdo	Centrarse en la colaboración y el consenso.
Lucha-vuelo	Intentar limitar el daño y reparar los daños colaterales.
Problema a resolver	Reunir y apoyar al equipo para una discusión.
Situaciones intratables	Separar a los dos equipos con diplomacia e intentar resolver las diferencias.

12 ¿En qué orden debería realizar las 3 historias a continuación para minimizar el costo de oportunidad?

Historia	Duración (días)	Valor diario una vez completada
A	1	$ 2.000
B	3	$ 10.000
C	5	$ 15.000

13 Coloque el número de técnica que corresponde a cada definición.

	Técnica	#	Definición
1	Analogía		Construir un modelo para estimar como podrían impactar al problema distintos escenarios.
2	Depuración de escopeta		Crear un ambiente controlado para probar soluciones al problema.
3	Descomposición		Partir el problema complejo en varios problemas más simples y fáciles de resolver.
4	Prueba y error		Probar diferentes alternativas hasta que alguna resuelva el problema.
5	Sandboxing		Probar varias posibles soluciones de problemas al mismo tiempo.
6	Simulación		Utilizar soluciones que fueron exitosas para otros problemas similares.

14 Indique si los siguientes indicadores son “rezagado” o “líder”.

Indicador	Rezagado	Líder
# Accidentes laborales		
Rentabilidad del proyecto		
# Defectos escapados		
Tiempo de ciclo		
% de trabajadores que utilizan casco		
% de trabajadores que respetan las normas		
Valor comercial de cada historia		
Velocidad del equipo		
Horas de refactorización		
Horas de revisiones de calidad		
WIP		
Rendimiento del equipo		

15 Señale si los siguientes ítems corresponden a una variación común o especial.

Variación	Común	Especial
Deben gestionarse de manera proactiva con acciones de control oportunas		
Deben seguir en observación mediante un análisis de tendencias		
desviaciones significativas		
Impacto bajo en los resultados		
La velocidad del equipo de las últimas 10 iteraciones tuvo variaciones en la velocidad entre 20 y 30.		
La velocidad del equipo subió de 20 a 50 en una iteración.		
No requieren acciones de control		
ocurren una sola vez		
Problema sistemático de pequeñas desviaciones		

16 Según un diagrama de control, coloque los siguientes números en la tabla a continuación: 180, 190, 200, 210, 220, 230.

Concepto	Valor
Fuera de control	
LCL	
Límite tolerancia inferior	
Límite tolerancia superior	
Media	
UCL	

17 Conteste verdadero o falso con relación al seguimiento y control de problemas.

Frase	V	F
El análisis de la variación es más importante que una tendencia		
Se puede disminuir el tiempo de ciclo bajando el WIP		
La verificación sirve para confirmar si estamos desarrollando el producto correcto		
La validación evalúa si el producto cumple las necesidades del cliente		
Los desarrolladores y probadores deben participar en la recopilación de requisitos		
Los probadores deberían formar parte del equipo de desarrollo desde el inicio		

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

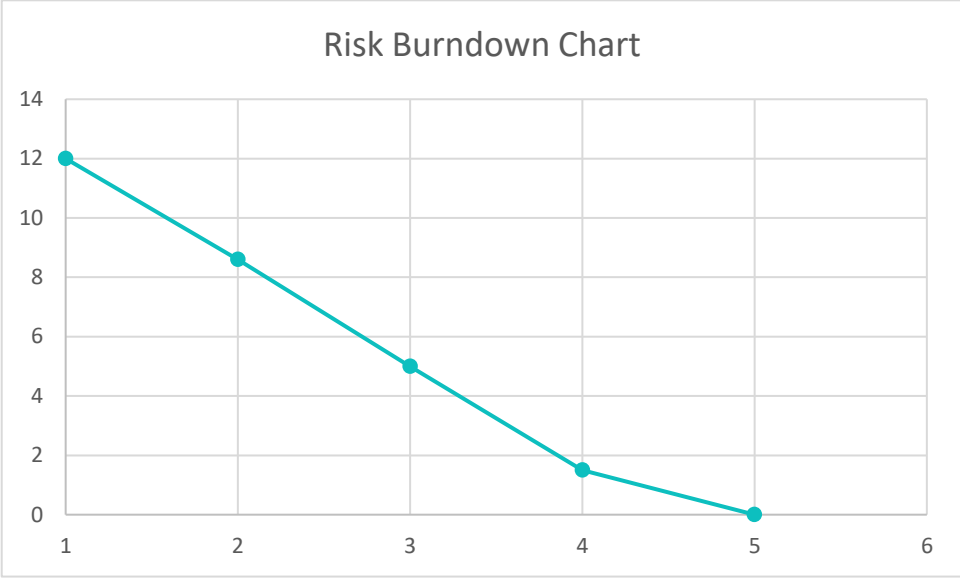
18 Para poder realizar el primer lanzamiento de un proyecto sobre tecnologías para el mantenimiento predictivo de una red ferroviaria, es necesario completar los cuatro requisitos solicitados por el cliente: A, B, C, D. El valor comercial estimado para cada requisito es: A: $120, B; $30, C: $430, D: $230. Por su parte, se ha realizado un análisis de riesgo cuantificando el valor esperado de los riesgos negativos de cada requisito en: A: $110; B: $100; C:$ 410; D: $200. ¿Qué recomendaría para este proyecto?

A. Realizar un ápice (spike) para evaluar si se puede reducir el riesgo del requisito B
B. Comenzar lo antes posible a desarrollar el requisito D
C. Completar todos los requisitos, excepto el B
D. Comenzar en la próxima iteración con el requisito C

19 Un equipo de proyecto realiza refactorización continua en cada iteración. ¿Cuál será el principal objetivo de esa actividad?

A. Acortar la vida útil de la inversión
B. Mantener baja la deuda técnica
C. Alterar la funcionalidad esperada del producto bajo desarrollo
D. Definir la severidad del riesgo

20 ¿Qué podríamos decir del diagrama de riesgo pendiente que se presenta a continuación?



A. El eje de ordenadas mide el impacto monetario del riesgo
B. A medida que avanza el proyecto, aparecen nuevos riesgos
C. En la iteración 3, el riesgo total asciende a 15 puntos (3 probabilidad x 5 impacto)
D. La exposición total del riesgo en la iteración 3 fue de 5 días de retraso

6.6 Respuesta ejercicios - Problemas

Riesgos

Riesgo	
Corrupción del alcance	√
Defecto del cronograma	√
Descompostura de la especificación	√
Financiación	
Negocios	
Pérdida de personal	√
Políticos	
Uso de tecnología	
Variación de la productividad	√

PESTLE

Tipo de riesgo	
P	Políticos
E	Económicos
S	Sociales
T	Tecnológicos
L	Legales
E	Ecológicos

Gestión de riesgos

Pasos	
2º	Analizar
1º	Identificar
4º	Monitorear
3º	Planificar respuesta

Severidad del riesgo

Riesgo	Probabilidad	Impacto	Severidad
A	Baja	Bajo	Baja
B	Baja	Alto	Media
C	Alta	Bajo	Media
D	Alta	Alto	Alta

Exposición del riesgo

Riesgo	Probabilidad	Impacto	Exposición
A	10%	$5.000	$500
B	20%	$80.000	$16.000
C	70%	$3.000	$2.100
D	80%	$60.000	$48.000
		Total	**$66.600**

Prioridad de historias de usuario

Historia	Riesgo	Valor	Prioridad
Alfa	Alto	Bajo	Baja
Beta	Alto	Alto	Media
Gama	Bajo	Bajo	Media
Delta	Bajo	Alto	Alta

7R Exposición del riesgo

Iteración	Riesgo	Probabilidad	Impacto (Días de retraso)	Exposición	Exposición total
1	a	10%	20	2	12
	b	20%	30	6	
	c	40%	10	4	
2	a	8%	20	1,6	7,1
	b	20%	20	4	
	c	30%	5	1,5	

Al finalizar la iteración # 2 el riesgo es de 7,1.

8R Detectar problemas

Frase	V	F
Con la intimidad cognitiva se puede sentir y anticipar malentendidos a nivel individual y grupal.	√	
El ratio error-retroalimentación es el número de nuevos defectos que tendrá el proyecto para corregir defectos existentes.	√	
La deuda técnica refleja el costo del retrabajo a realizar hoy para mitigar futuros problemas.		√
Los 7 ¿Por qué? es una técnica para explorar las relaciones de causa y efecto subyacentes a un problema particular para detectar la causa-raíz de ese problema.		√
Los defectos escapados son los más costosos de corregir.	√	
Los defectos escapados son un indicador del nivel de calidad del proyecto.	√	
Se gasta más dinero cuando el equipo de desarrollo detecta el problema antes que el usuario final.		√
Un defecto escapado es aquel que fue descubierto por el equipos de prueba.		√
Una herramienta para detectar problemas es realizando chequeos sobre la salud del equipo.	√	

9R Abismo de ejecución y evaluación

Abismo	Ejecución	Evaluación
Distancia entre la respuesta del sistema y la interpretación del usuario.		√
Distancia entre lo que quiere hacer el usuario y los medios para lograrlo al interactuar con el sistema.	√	
El usuario es incapaz de interpretar si la acción que realizó fue exitosa o no.		√
El usuario sabe qué objetivos quiere lograr, pero desconoce qué pasos o acciones realizar para llevarlos a cabo.	√	
El usuario tiene dudas si está realizando el proceso correcto para utilizar un producto.		√
Un usuario tiene dudas sobre los pasos a realizar para interactuar con un producto.	√	

10R Refactorización

Ítem	
Cada vez que aparezca un problema, el coach ágil debe reunir a todo el equipo para compartir el problema y analizar alternativas de solución.	
Eliminar el código innecesario para facilitar el mantenimiento en el futuro.	√
En la cultura organizacional ágil se crea un ambiente dónde se aceptan los fracasos para la mejora continua.	
Limpiar el código simplificando el código fuente sin cambiar su comportamiento.	√
Limpieza, mantenimiento y estandarización del código en cada iteración, para mantener baja la deuda técnica.	√
Los miembros del equipo son los que están mejor preparados para encontrar soluciones a los problemas que aparezcan durante el desarrollo.	
Mejorar el diseño de un producto mejorando su disposición para el mantenimiento, sin alterar la funcionalidad esperada.	√
Mejorar la operación interna de un componente sin cambiar su comportamiento externo.	√

11R Niveles de conflicto

Nivel de conflicto	Resolución
3 Contienda o combate	Asegurarse de que las rivalidades personales pasen a un segundo plano y todos negocien una discusión abierta.
2 Desacuerdo	Centrarse en la colaboración y el consenso.
4 Lucha-vuelo	Intentar limitar el daño y reparar los daños colaterales.
1 Problema a resolver	Reunir y apoyar al equipo para una discusión.
5 Situaciones intratables	Separar a los dos equipos con diplomacia e intentar resolver las diferencias.

12R CD3

Costo sin priorización: $177.000

Historia	Duración (días)	Valor diario una vez completada	CD3	Demora	Costo oportunidad
A	1	$ 2.000	$ 2.000	1	$ 2.000
B	3	$ 10.000	$ 3.333	4	$ 40.000
C	5	$ 15.000	$ 3.000	9	$ 135.000
TOTAL	**9**				**$ 177.000**

Costo priorizando por CD3: $168.000

Historia	Duración (días)	Valor diario una vez completada	CD3	Demora	Costo oportunidad
B	3	$ 10.000	$ 3.333	3	$ 30.000
C	5	$ 15.000	$ 3.000	8	$ 120.000
A	1	$ 2.000	$ 2.000	9	$ 18.000
TOTAL	**9**				**$ 168.000**

13R Técnicas para resolución de problemas

	Técnica	#	Definición
1	Analogía	6	Construir un modelo para estimar como podrían impactar al problema distintos escenarios.
2	Depuración de escopeta	5	Crear un ambiente controlado para probar soluciones al problema.
3	Descomposición	3	Partir el problema complejo en varios problemas más simples y fáciles de resolver.
4	Prueba y error	4	Probar diferentes alternativas hasta que alguna resuelva el problema.
5	Sandboxing	2	Probar varias posibles soluciones de problemas al mismo tiempo.
6	Simulación	1	Utilizar soluciones que fueron exitosas para otros problemas similares.

14R Indicadores

Indicador rezagado - Objetivo	Indicadores líderes
Accidentes laborales	% de trabajadores que utilizan casco % de trabajadores que respetan las normas
Rentabilidad del proyecto	Valor comercial de cada historia Velocidad del equipo
Defectos escapados	Horas de refactorización Horas de revisiones de calidad
Tiempo de ciclo	WIP Rendimiento del equipo

15R Tipos de variación

Variación	Común	Especial
Deben gestionarse de manera proactiva con acciones de control oportunas		√
Deben seguir en observación mediante un análisis de tendencias	√	
desviaciones significativas		√
Impacto bajo en los resultados	√	
La velocidad del equipo de las últimas 10 iteraciones tuvo variaciones en la velocidad entre 20 y 30.	√	
La velocidad del equipo subió de 20 a 50 en una iteración.		√
No requieren acciones de control	√	
ocurren una sola vez		√
Problema sistemático de pequeñas desviaciones	√	

16R Diagrama de control

Concepto	Valor
Fuera de control	230
LCL	180
Límite tolerancia inferior	190
Límite tolerancia superior	210
Media	200
UCL	220

17R Seguimiento y control

Frase	V	F
El análisis de la variación es más importante que una tendencia		√
Se puede disminuir el tiempo de ciclo bajando el WIP	√	
La verificación sirve para confirmar si estamos desarrollando el producto correcto		√
La validación evalúa si el producto cumple las necesidades del cliente	√	
Los desarrolladores y probadores deben participar en la recopilación de requisitos	√	
Los probadores deberían formar parte del equipo de desarrollo desde el inicio	√	

18R Respuesta A. Dado el contexto de la pregunta deberíamos suponer que no es posible un lanzamiento parcial sin completar los cuatro requisitos que ha solicitado el cliente. Si estimamos el valor comercial neto de los riesgos para cada requisito, tenemos: A: $120 - $110 = $10; B: $30 - $100 = -$70; C: $430 - $410 = $20; D: $230 - $200 = $30. El requisito B tiene un valor negativo de $70 que no alcanzará a cubrir el valor positivo de los otros tres requisitos $60 ($10 + $20 + $30). Por lo tanto, no deberíamos avanzar con este proyecto hasta no realizar un análisis de riesgo más profundo para evaluar si el proyecto como un todo será rentable. / Los spikes o ápices son un tipo de historia de usuario utilizado para obtener el conocimiento necesario para reducir el riesgo de un enfoque técnico, comprender mejor un requisito o aumentar la confiabilidad de una estimación de la historia.

19R Respuesta B. Refactorización: mejorar el diseño de un producto mejorando su disposición para el mantenimiento y otros atributos deseados, sin alterar la funcionalidad esperada. En otras palabras, mejorar la operación interna de un componente sin cambiar su comportamiento externo. Con la refactorización continua (limpieza, mantenimiento y estandarización) dentro de cada iteración, se mantiene baja la deuda técnica, por lo que se puede extender la vida útil de la inversión de una empresa en activos de software y los usuarios pueden continuar experimentando un flujo de valor en los años venideros.

20R Respuesta D. En el diagrama de riesgos pendientes se grafica la exposición total del riesgo en las ordenadas y el número de iteración en la abscisa. Si bien pueden aparecer riesgos a lo largo de todo el proyecto, por lo general, a medida que avanzan las iteraciones se van completado historias del proyecto y los riesgos tienden a disminuir. En la iteración 3, la exposición total de riesgos del proyecto fue de 5 días.

Capítulo 7

Mejora continua

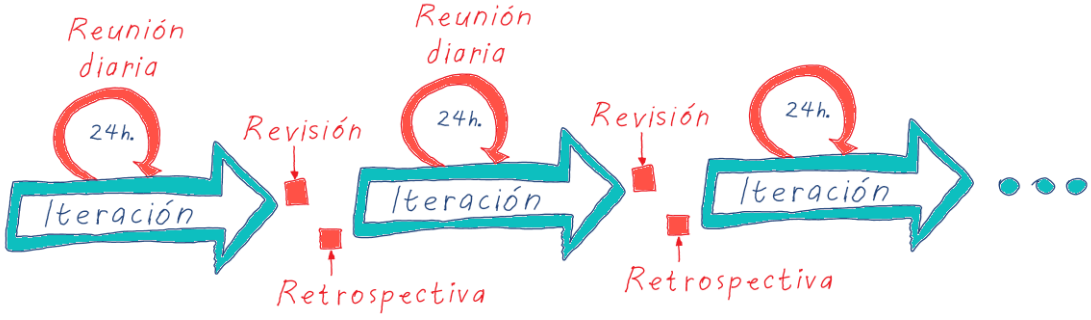
7.1 Retroalimentación

Los bucles de retroalimentación son mecanismos que se utilizan para validar y obtener retroalimentación negativa y positiva de manera inmediata sobre el proceso de desarrollo del producto. Esta retroalimentación es la base para la mejora continua.

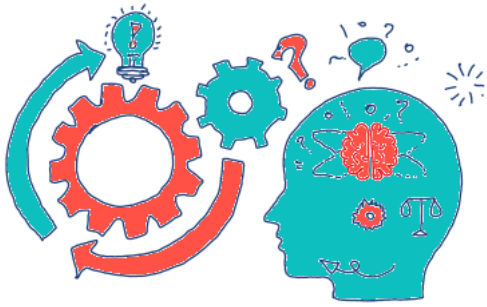
Para el éxito de un proyecto ágil, es fundamental crear una cultura organizacional para implementar pequeñas mejoras a lo largo de todo el proyecto.

El proceso ágil tiene puntos de control integrados para facilitar la retroalimentación y la colaboración mediante reuniones tales como:

1. Diaria: permite a los miembros del equipo compartir actualizaciones de estado e identificar los obstáculos que se interponen en su camino.
2. Revisión de la iteración: se presenta el incremento del producto al dueño del producto y otros interesados clave para evaluar el proyecto y comentar sobre las necesidades aún no satisfechas.
3. Retrospectiva: permite al equipo mirar hacia atrás en lo que salió bien y lo que podría hacerse mejor.



Ciclo de aprendizaje

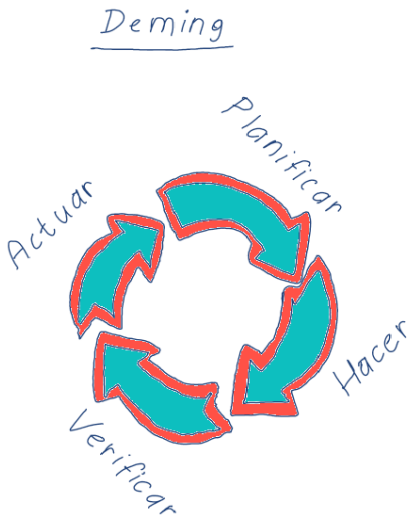


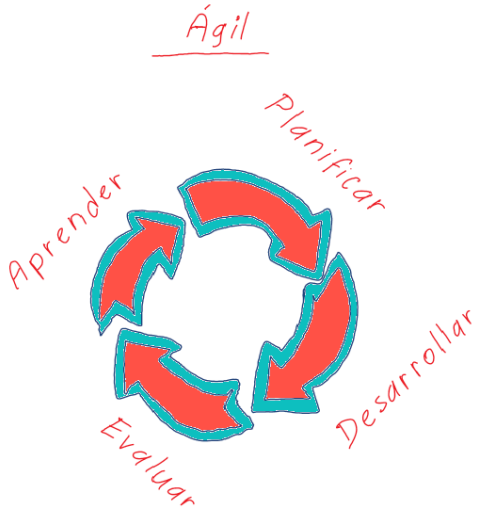
En el desarrollo de proyectos ágiles se suele comenzar con un bajo conocimiento de cómo será el producto final, por lo que es necesario un aprendizaje permanente para ir mejorando el desarrollo del producto para entregar el máximo valor al cliente.

Este aprendizaje continuo sobre los requisitos del cliente y los cambios tecnológicos son los que justifican mantener ciclos cortos de aprendizaje en cada iteración para poder entregar un producto viable.

Al mantener ciclos de aprendizaje cortos, los nuevos conocimientos que se adquieren en cada iteración se pueden incorporar rápidamente al proyecto.

Este ciclo de aprendizaje se puede representar como “Planificar-Desarrollar-Evaluar-Aprender”, que se asemeja al ciclo tradicional de Deming “Planificar-Hacer-Verificar-Actuar” (PDCA: Plan-Do-Check-Act)





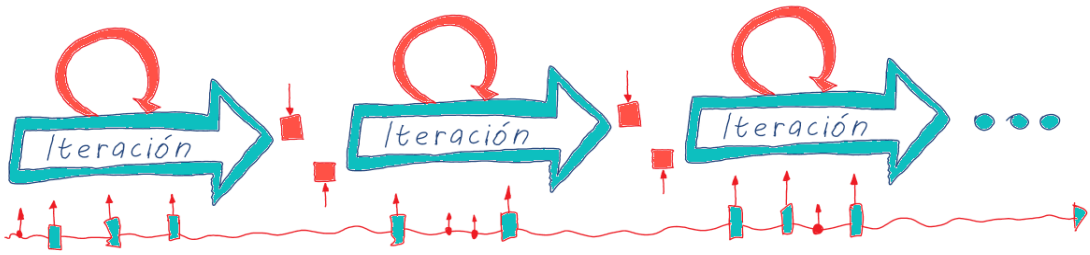
Retroalimentación continua

Si bien el proceso ágil acorta los ciclos de retroalimentación con relación a procesos de ciclos predictivos, algunos proyectos ágiles necesitan retroalimentación y colaboración aun más frecuentes.

Por ejemplo, hay proyectos que están utilizando aplicaciones de bajo código (low-code) para que el cliente o usuario pueda ofrecer retroalimentación inmediata a los desarrolladores a lo largo de todo el proyecto, sin necesidad de esperar a que finalice una iteración.

Estos modelos visuales de retroalimentación de bajo código son fáciles de utilizar, por lo que facilitan la colaboración frecuente y continua entre los desarrolladores y aquellos interesados de bajo conocimiento técnico.

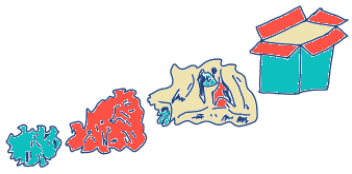
En cualquier momento, incluso al principio del proceso de desarrollo, el cliente y desarrollador pueden sentarse juntos para discutir y revisar la funcionalidad para recopilar comentarios, validar supuestos e identificar mejoras.



El valor de una plataforma low-code es que permite el desarrollo iterativo y colaborativo, innovando sin detenerse por los detalles técnicos.

7.2 Mejorar los procesos

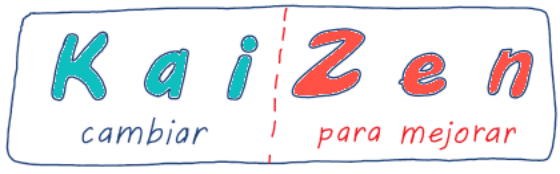
Las organizaciones ágiles han creado una cultura donde se permite fracasar para poder aprender rápidamente de esos errores y de esa forma promover la mejora continua.



Mientras antes se fracase, más barato será el costo de ese aprendizaje y más rápido se podrá mejorar el proyecto y sus procesos para aprender de los errores.

Kaizen

Kaizen es una palabra japonesa que suele ser sinónimo de "mejora continua". Parte del concepto de que todo proceso, producto o servicio puede mejorarse.



Un lema del Kaizen es «*hoy mejor que ayer y mañana mejor que hoy*»; o sea, todos los días se pueden hacer mejor las cosas.

Las "5 S para implementar Kaizen":

- Seiri (Clasificación): eliminar lo innecesario del espacio de trabajo.
- Seiton (Orden): organizar el espacio de trabajo de forma eficaz.
- Seisō (Limpieza): eliminar la suciedad del lugar de trabajo.
- Seiketsu (Normalización): señalizar anomalías para prevenir la suciedad y el desorden.
- Shitsuke (Mantener la disciplina): seguir mejorando para la mejora continua.

El objetivo de la técnica 5 S es eliminar los desperdicios del espacio de trabajo, mantenerlo limpio periódicamente y seguir procesos estandarizados para facilitar un flujo fluido.

Kaizen aboga por la participación de las personas en todos los niveles para que todos presenten pequeñas mejoras de forma continua.

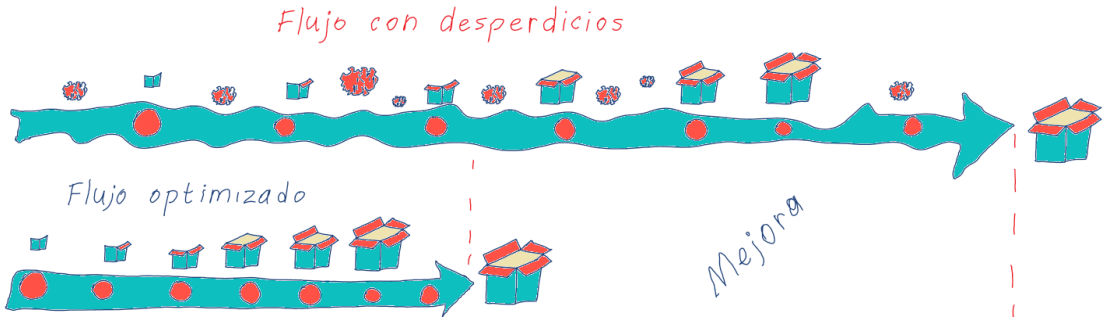
Los grandes resultados provienen de pequeños cambios acumulados en el tiempo.

Estos pequeños cambios mejoran la productividad, la eficacia y la innovación al tiempo que reducen el desperdicio.

Mapeo de flujo del valor (Value stream mapping)

El mapeo del flujo de valor es un diagrama de flujo de información para ilustrar, analizar y mejorar los pasos necesarios para entregar un producto o servicio desde su desarrollo hasta la entrega al cliente.

Este mapa es un concepto lean que se utiliza para encontrar y eliminar desperdicios, obteniendo mejoras continuas de procesos optimizados. Por lo general, se dibuja primero un mapa del estado actual y luego se modela lo que sería el mapa futuro o ideal.



Pasos

Crear un mapa de flujo de valor del proceso actual identificando los pasos, colas, retrasos y flujos de información.

Revisar el mapa para encontrar retrasos, desperdicios y restricciones.

Crear un nuevo mapa de flujo de valor del estado optimizado que se logrará en el futuro, eliminando retrasos, desperdicios y restricciones.

Desarrollar una hoja de ruta para lograr el estado futuro.

Planificar la revisión del proceso para calibrar y optimizar continuamente.

Análisis de procesos: mejora de la eficiencia y la eficacia de los procesos mediante la identificación y eliminación de gastos innecesarios, restricciones o cualquier cosa que no agregue valor al cliente.

Reunión de proyecto pre-mortem

Durante una reunión de proyecto pre-mortem el equipo imagina que el proyecto ha fracasado y luego trabaja hacia atrás para determinar cuáles podrían haber sido las causas de ese fracaso.

Esto sería la lógica opuesta a la reunión de retrospectiva que se lleva a cabo post iteración donde se aprende de los fracasos una vez que ya ocurrieron.

El objetivo de la reunión pre-mortem es anticipar los riesgos negativos y actuar para que no lleguen a producirse. Los riesgos se evalúan incluyendo las causas, probabilidades de ocurrencia e impactos y se elabora un plan de respuesta para los riesgos más significativos.

Esta técnica se lleva a cabo en equipo con juego de roles, en el que el líder comunica al resto del equipo que el proyecto ha fracasado para que realicen un listado de las causas que llevaron el negocio a su muerte.

Esta situación ficticia de fracaso de proyecto facilita al equipo la identificación de riesgos en las fases iniciales del proyecto.

Pasos

El facilitador anuncia al equipo que el proyecto ha fracasado.

Se pregunta al equipo: *¿Por qué fracasó el proyecto?*

Tormenta de ideas donde los participantes explican posibles causas del fracaso.

Se recopilan las potenciales causas de fracaso y sus riesgos asociados para mejorar la planificación del proyecto.

Control empírico del proceso

En los proyectos ágiles las decisiones se toman en base a la observación y la experimentación, en lugar de una planificación inicial detallada.

El control empírico del proceso se basa en tres conceptos:

Transparencia: cualquier persona tiene acceso a los artefactos, reuniones y radiadores de información.

Inspección: utilización de radiadores de información, recopilación de comentarios del cliente, aprobación de los entregables por parte del dueño del producto.

Adaptación: el equipo realiza mejoras continuas en el trabajo que están realizando.

Kanban Kata

Kanban Kata es otra técnica de mejora continua que utiliza una serie de preguntas para ayudar a mejorar en pequeños pasos, de modo que el trabajo diario y las mejoras sucedan simultáneamente.

Por ejemplo:

P1: ¿Cuál es el objetivo que queremos lograr?

P2: ¿Cuál es el estado actual de ese objetivo?

P3: ¿Cuáles son los impedimentos o bloqueos en el camino?

P4: ¿Cuál es el siguiente paso para resolver el impedimento?

P5: ¿Cuándo podemos ver lo que aprendimos al dar ese paso?

Procesos a medida (Tailoring processes)

Para la adaptación de cualquier marco o proceso ágil primero deberíamos comprender las razones detrás de una práctica específica que se incluye en el marco, antes de decidir que no la necesitamos.

Eliminar partes de un marco sin comprender su propósito es como tirar una pieza que no nos gusta de una máquina sin saber lo que hace y luego esperar que esta máquina siga funcionando correctamente.

Una vez que el equipo comprende los procesos de un marco ágil, puede explorar diferentes formas de ajustar los procesos a las necesidades de su organización sin necesidad de reinventar la rueda.

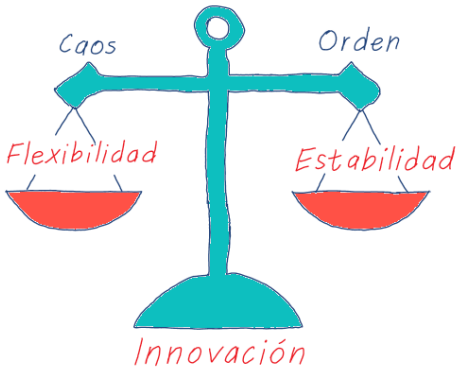
Hay algunas metodologías como Kanban que son sumamente flexibles para adaptar y optimizar los procesos a la medida de cada organización, de cada proyecto y de cada situación en particular. Kanban reconoce que cada situación es única, por lo que se pueden adaptar los procesos para cada proyecto. Un espíritu similar se encuentra en el marco de Disciplined Agile.

Por su parte, metodologías como Scrum son menos amigables para adaptar los procesos. La metodología Scrum tiene recomendaciones para transformar toda la organización a un ambiente ágil, pero respetando siempre sus artefactos y eventos.

En la tabla a continuación se resumen procesos en función de la incertidumbre de los requisitos del cliente y la incertidumbre tecnológica.

Incertidumbre		Marco / Procesos
Requisitos	Tecnológica	
Alta	Alta	Caos. Ningún proceso va a funcionar.
Media	Media	Adaptativo. Procesos ágiles.
Baja	Baja	Predictivo. Procesos en cascada.

Innovación continua



La agilidad es la capacidad de crear y responder a los cambios para obtener ganancias en un entorno empresarial turbulento, equilibrando la estabilidad con la flexibilidad.

Mientras que la falta de estructura crea caos e ineficiencia, demasiada estructura crea rigidez y falta de innovación. La innovación ocurre más fácilmente en el punto de equilibrio entre el caos y el orden, entre la flexibilidad y la estabilidad.

Las culturas de innovación aprenden realizando experimentos diseñados para progresar iterativamente hacia una meta. Estos experimentos no fallan, sino que producen los datos necesarios para aceptar o rechazar una hipótesis.

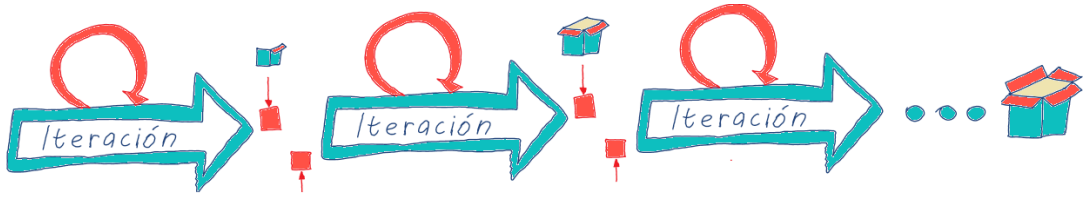
Hay empresas que no innovan lo suficiente o paralizan sus innovaciones debido a una cultura sobre el miedo al fracaso. Las culturas de innovación se basan en aprender de los experimentos e incorporar esos conocimientos en la exploración futura.

Cuando los líderes crean la seguridad psicológica de que es posible fracasar, se alienta a las personas a experimentar dentro de ciertos límites. Los miembros del equipo tienen autoridad para resolver grandes problemas, aprovechar oportunidades y hacerlo sin temor a ser culpables, incluso cuando los resultados de los experimentos no sean los esperados. [29]

[29] © Scaled Agile, Inc.

7.3 Mejorar el producto

El desarrollo iterativo e incremental con una retroalimentación constante del cliente es una forma para la mejora continua del producto.



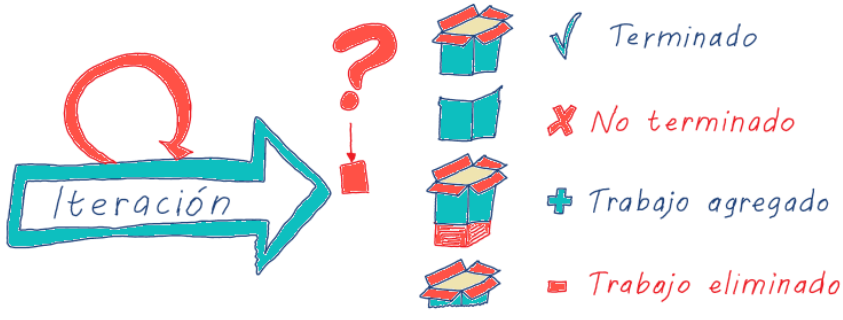
Mediante este ciclo de desarrollar pequeños incrementos, revisarlos, discutir como mejorarlos e implementar pequeñas mejoras, el producto o servicio se va desarrollando a través del proceso de mejor continua.

Revisión de la iteración (Sprint review)

La revisión de la iteración es una reunión que se realiza al final de cada iteración, donde el equipo de desarrollo presenta al dueño del producto el incremento completado para obtener comentarios sobre el producto desarrollado.

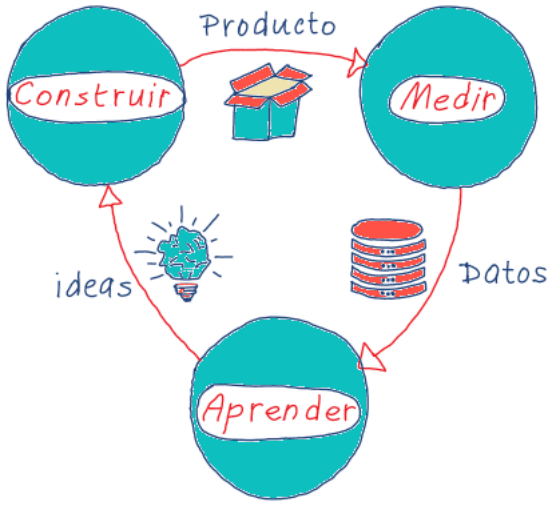
Durante esta reunión, el dueño del producto, quien representa los intereses del cliente, revisa si el trabajo realizado cumple con el objetivo de la iteración y con la definición de terminado.

Algunos clientes solicitan una demostración del producto para probarlo, antes de aceptar si cumple o no con la definición de terminado. Como resultado de esta revisión, el dueño del producto determina el estado del trabajo implementado en la iteración:



El resultado final de esta reunión es una revisión del product backlog para priorizar y planificar las actividades de la próxima iteración. También se actualiza el plan de lanzamiento cuando sea necesario.

Ciclo de retroalimentación del producto (Product feedback loop)



La actividad fundamental de una startup es convertir ideas en productos, medir cómo responden los clientes y luego aprender si pivotar o perseverar. Esto sirve como base para el marco "construir-medir-aprender". [30]

Construir una idea en algo tangible que los clientes deberían amar.

Medir si a los clientes les encanta el producto.

Aprender de los errores para crear un producto que a los clientes les encantará.

El ciclo de retroalimentación del producto es el proceso de recopilar continuamente los comentarios de los clientes y mejorar el producto en función de sus opiniones.

Desarrollar un producto para presentar a los usuarios y obtener una retroalimentación temprano lo antes posible. Por ejemplo: folleto, mockup, prototipo, producto viable mínimo, etc.

Evaluar si a los clientes les gusta el producto a través de: entrevistas, encuestas, departamento de atención al cliente, métricas de usabilidad, ventas, redes sociales, etc.

Entre algunas de las **encuestas** más utilizadas figuran: NPS (net promoter score), CSAT (customer satisfaction) y CES (customer effort score).

[30] Eric Ries, Lean Startup.

NPS (Net promoter score)

La encuesta de puntuación del promotor neto (NPS) mide la lealtad de los clientes basado en una sola pregunta:

¿Qué tan probable es que recomiende este producto a un amigo o colega?

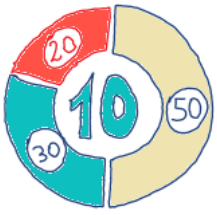
Los participantes responden en una escala de 0 a 10, donde 0 es "Muy improbable" y 10 es "Definitivamente lo recomendaría".

Promotores: responden 9 o 10. Es muy probable que recomienden su producto.

Pasivos: responden 7 u 8. Están satisfechos con el producto, pero no son amantes.

Detractores: responden entre 0 y 6. No están satisfechos y podrían hablar mal.

Para obtener el NPS se restan los detractores de los promotores y se consigue un porcentaje. El índice NPS puede ser tan bajo como -100% (todo el mundo es un detractor) o tan alto como 100% (todo el mundo es un promotor).



Ejemplo de cálculo. Supongamos que la encuesta fue realizada a 1.000 personas con los siguientes resultados: 300 personas calificaron 9 o 10, 500 personas calificaron 7 u 8 y 200 personas calificaron entre 0 y 6. En este caso tenemos 30% de promotores (300 / 1000) y 20% de detractores (200 /1000), por lo que el NPS es de 10% (30% - 20%).

Un NPS superior a 0% es favorable y un NPS superior al 50% es excelente.

Identificando a los promotores sabremos a dónde invertir los recursos de comercialización. Por su parte, identificando a los detractores podemos invertir en acciones proactivas para mejorar el producto de tal forma que agregue valor.

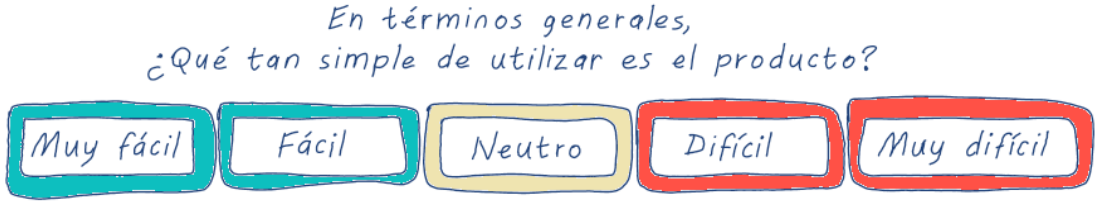
CSAT (Customer Satisfaction Score)

La encuesta de puntuación de satisfacción del cliente (CSAT) se utiliza para medir de manera simple la satisfacción de experiencias en una nueva funcionalidad.

Esta encuesta se suele utilizar para medir la satisfacción del cliente después de que un usuario activa la funcionalidad, antes de que el cliente renueve el producto o después de las sesiones de soporte al cliente.

CES (Customer Effort Score)

La encuesta de puntuación sobre el esfuerzo del cliente (CES) se utiliza para averiguar si el producto es fácil de usar.



El mejor momento para solicitar la respuesta del usuario sería después de:

- El primer uso de una nueva funcionalidad
- Que los usuarios alcancen un hito o aprendan a utilizar el producto
- Ponerse en contacto con el servicio de atención al cliente
- Una interacción con el producto que dio lugar al pago, suscripción o actualización.

Reflexionar sobre los datos obtenidos para priorizarlos y convertirlos en información que sea de utilidad para mejorar el producto. Mejore continuamente la forma de capturar los comentarios sobre los productos.

Demostración del sistema (system demo)

En la metodología SAFe, cuando hay varios equipos ágiles desarrollando diferentes componentes de un producto incremental, al terminar la iteración se lleva a cabo una demostración del sistema para presentar a los interesados todos esos componentes integrados.

La mejor medida real de valor, velocidad y progreso del proyecto es la demostración del trabajo totalmente integrado de todos los equipos.

La demostración del sistema es presentada en conjunto por los gerentes del producto y dueños del producto de cada equipo. Los principales interesados de esta demostración son los dueños del negocio, patrocinadores, clientes y otros interesados clave.

Cada demostración brinda a los interesados una medida objetiva del progreso de desarrollo del producto para que puedan proveer una retroalimentación inmediata.

El objetivo de la demostración del sistema es aprender de la experiencia de desarrollo más reciente para ajustar el curso de acción de las próximas iteraciones.

7.4 Mejorar el equipo

La mejora continua de los miembros del equipo será fundamental para poder mejorar los procesos y el producto.

Las metodologías ágiles incluyen momentos de reflexión a lo largo del proyecto para evaluar el rendimiento de las personas con el objetivo de mejorar su desarrollo individual y colectivo.

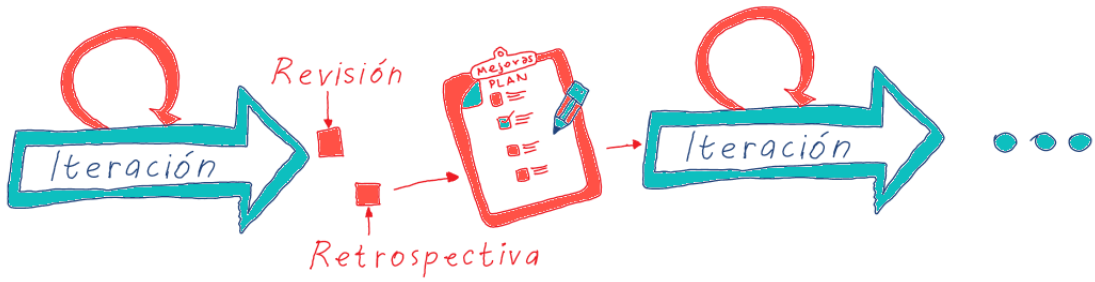
La motivación permanente del equipo es clave para la mejora continua. Algunos factores para aumentar la motivación de un equipo ágil suelen ser:

- Reducir el tamaño del equipo tanto como sea posible
- Empoderar al equipo para que ellos tomen sus propias decisiones
- Conectar al equipo con los usuarios
- Permitir que el equipo sea dueño de su propio proceso
- Brindar al equipo una porción del sistema donde puedan ser dueños de la definición, entrega y despliegue de valor

Las reuniones de retrospectiva son un evento muy importante para que el equipo pueda compartir el conocimiento y pueda realizar auto evaluaciones de manera frecuente.

Retrospectiva de la iteración (sprint retrospective)

La reunión de retrospectiva de la iteración es una oportunidad para el equipo de inspeccionarse a sí mismo y crear un plan de mejora que se pondrá en marcha en la siguiente iteración.



El objetivo de esta reunión es mejorar la productividad, las habilidades del equipo y la calidad del producto. En este evento participa todo el equipo y se realiza después de la reunión de revisión de la iteración.

Las reuniones de retrospectiva permiten al equipo mirar hacia atrás para reflexionar en lo que salió bien y lo que podría hacerse mejor para seguir mejorando en las próximas iteraciones.

Como resultado de la reunión de retrospectiva se obtiene:

- ☑ Plan de mejora del proceso
- ☑ Mejora en la calidad del producto
- ☑ Mejora en la productividad del equipo

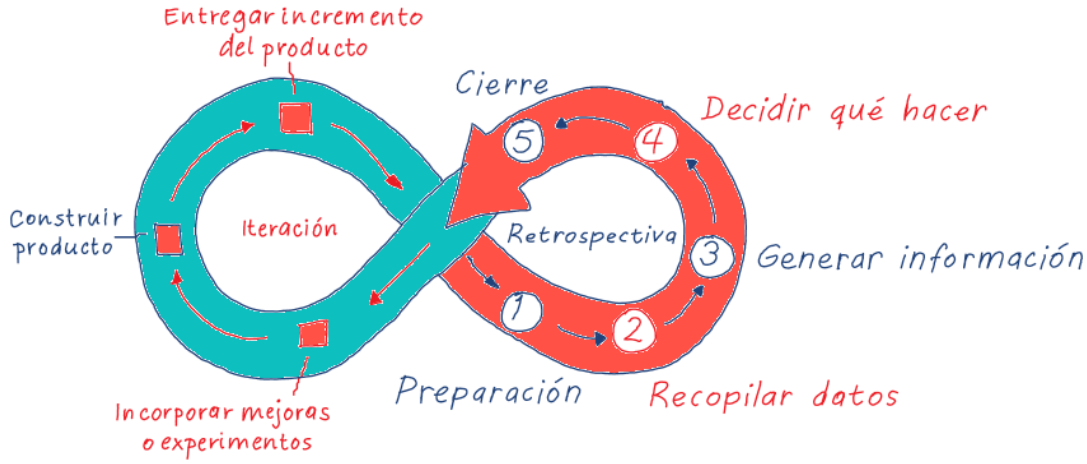
Además de las retrospectivas de la iteración que es la más común, también pueden realizarse retrospectivas más largas e intensivas en hitos importantes del proyecto. Por ejemplo:

- Retrospectivas de lanzamiento
- Retrospectivas sorpresa (realizadas cuando un evento inesperado cambia la situación)
- Retrospectivas de fin del proyecto

Estas retrospectivas más intensivas, brindan la oportunidad de reflexionar más profundamente sobre las experiencias y condensar lecciones clave para compartir con el resto de la organización.

Si el equipo quisiera aumentar la tasa de mejora de procesos se podrían llevar a cabo retrospectivas más seguido para identificar mejoras.

Fases de la retrospectiva



① Preparación / El coach ágil en conjunto con el equipo, definen las reglas básicas para garantizar que todos se sientan cómodos discutiendo los problemas del proyecto.

② Recopilar datos / El equipo recopila datos de los problemas que se enfrentan durante la iteración mediante diagrama de espina de pescado, 5 porqués, gráficos de control, etc.

③ Generar información / Basándose en los datos recopilados, el equipo puede inferir la causa raíz de los problemas que se enfrentan, transformando esos datos en información.

④ Decidir qué hacer / Basándose en la información, el equipo decide los planes de mejora que se implementarán en las próximas iteraciones para no cometer los mismos problemas. Algunas herramientas utilizadas para estas decisiones suelen ser:

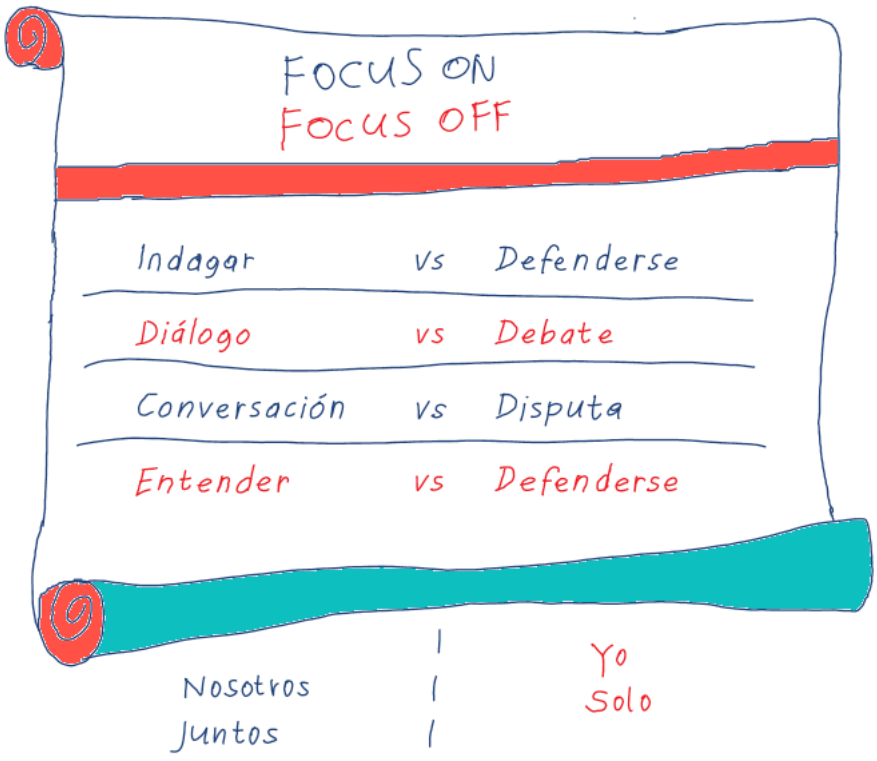
- Igual que - más de - menos de (SAMOLO: Same as – more of – less of)
  = Igual que: prácticas que el equipo considera valiosas y que seguiría utilizando.
  + Más de: prácticas valiosas, pero el equipo siente que no están haciendo lo suficiente.
  - Menos de: prácticas en las que el equipo encuentra poco valor y les gustaría desalentar.
- Empezar a hacer - dejar de hacer - seguir haciendo (Start – Stop – Keep)
  Empezar a hacer: actividades no realizadas, pero hay que hacerlas porque agregarán valor.
  Dejar de hacer: actividades que no agregan valor y deben detenerse.
  Seguir haciendo: actividades que agregan valor y vale la pena continuar.

⑤ Cierre / El coach ágil agradece al equipo por su contribución y los miembros del equipo agradecen a sus colegas por su ayuda para resolver problemas durante la iteración.

Focus on / focus off

Como parte de la preparación de la reunión de retrospectiva se puede utilizar el ejercicio de "Focus on - focus off" creando un ambiente de colaboración para que los miembros del equipo se enfoquen en los patrones productivos: indagar, diálogo, conversación, entender al otro.

Se divide en grupos al equipo para que comparen pares de palabras y discutan qué significan y cómo impactan en el trabajo en equipo.



Una vez que los participantes comparten las ventajas y desventajas de esos pares de palabras, se pueden utilizar esos comentarios para establecer acuerdos de trabajo para la reunión.

Establecer el escenario con una mentalidad para la comunicación productiva ayuda a los participantes a dejar de lado el miedo a la culpa y el juicio.

ESVP (Explorador, Shopper, Vacacionista, Prisionero)

Otra técnica que se suele utilizar en la fase de preparación de la retrospectiva es ESVP:

Explorador: deseoso de descubrir nuevas ideas y aprender todo sobre el proyecto.

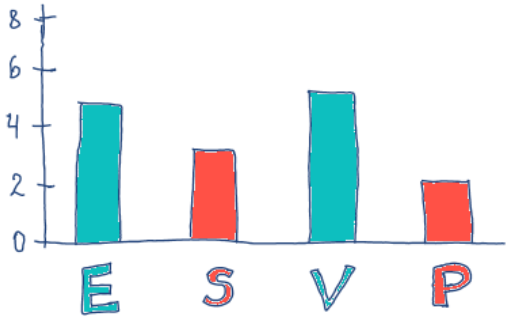
Shopper o Comprador: mira lo que aparece y si algo le gusta, se lo lleva para usar.

Vacacionista: no está interesado en la retrospectiva, pero contento de descansar y tomar café un rato.

Prisionero: es el obligado a participar, le gustaría estar haciendo otra cosa.

Pasos:

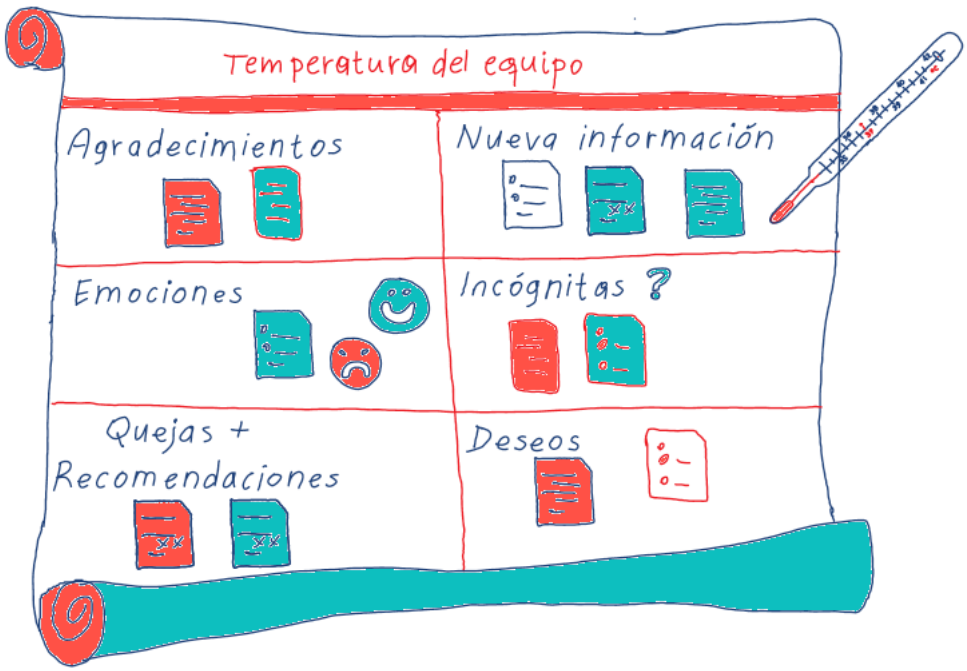
1. Los participantes de la reunión responden de manera anónima en un papel a cuál de estas 4 categorías ESVP pertenecen.
2. El facilitador, muestra los resultados en un histograma manteniendo siempre el anonimato de los participantes.
3. El facilitador pregunta a los participantes cómo se sienten con esos resultados y cómo podrían impactar en la retrospectiva.



Esta herramienta es de utilidad para generar debates enriquecedores a partir de la propia actitud y expectativa que el equipo demuestra sobre el proyecto.

Temperatura del equipo

Durante la reunión de retrospectiva se puede realizar una lectura sobre la temperatura de los miembros del equipo. Por ejemplo, en un rotafolio se colocan secciones con la retroalimentación de los miembros del equipo para ayudarlos a mantener o recuperar una mentalidad positiva.



Agradecimientos: cada miembro del equipo puede agradecer a sus colegas por el trabajo realizado durante la iteración.

Emociones: el equipo presenta los momentos de la iteración que fueron los más emocionantes y memorables para ellos.

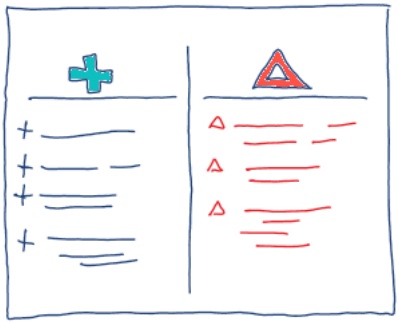
Quejas con Recomendaciones: cada queja debe ir seguida de una recomendación específica sobre cómo remediar la situación para intentar que no vuelva a suceder.

Nueva información: los miembros del equipo comparten información que los otros pueden desconocer o necesitar.

Enigmas (incógnitas): ¿Hay algún problema que no hayamos cerrado o que el equipo no haya podido responder?

Deseos: sugerencias y deseos de los miembros del equipo.

Plus - Delta



Una herramienta para generar comentarios constructivos al finalizar la retrospectiva es la herramienta Plus / Delta.

Pasos:

1. Haga dos columnas en una pizarra: una para "más" y otra para "delta" (el símbolo griego para el cambio).
2. Pida al equipo que reflexione sobre lo positivo o repetible de una actividad y capture sus pensamientos en la columna "+".
3. Pida al grupo que haga una lluvia de ideas sobre lo que cambiarían al respecto y anótelos en la columna “delta”.

Al enfocarse en el cambio en lugar de los negativos directos, será más probable que el grupo comparta su verdadera evaluación y al mismo tiempo genere ideas de mejora.

Retorno del tiempo invertido (ROTI: return on time invested)

ROTI es un método ágil, rápido y fácil para medir el tiempo dedicado a reuniones, talleres, presentaciones y conferencias.

Los miembros del equipo votan al final de la reunión de retrospectiva con los dedos de su mano sobre la efectividad de la reunión:

Un dedo: inútil, perdí el tiempo en esta reunión

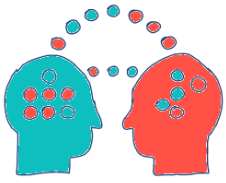
Dos dedos: útil, pero perdí un poco de tiempo al estar presente

Tres dedos: normal, no perdí tiempo, pero tampoco encontré demasiado valor

Cuatro dedos: buena, no perdí el tiempo por asistir

Cinco dedos: excelente, me habría arrepentido de no haber participado

Compartir el conocimiento

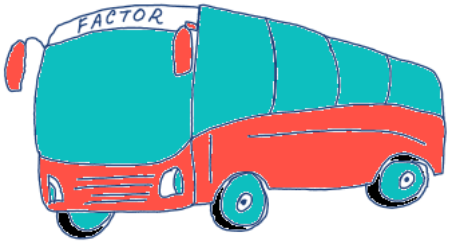


Compartir el conocimiento es fundamental para la mejora continua del equipo. En metodologías ágiles se propicia la comunicación y el conocimiento compartido mediante miembros del equipo con perfil "T" (expertos en su especialidad y a la vez interesados en colaborar en otras áreas). Este ambiente, con personas abiertas e interesadas, donde la comunicación entre todo el equipo es continua, facilita el aprendizaje y la integración de nuevas personas.

Guild o gremios

Grupos de personas transversales a la organización que se juntan para compartir conocimientos y buenas prácticas. Cada gremio tiene un coordinador que facilita los eventos. Un gremio a menudo incluye los grupos de interés y sus miembros que trabajan en una disciplina. Por ejemplo, el gremio de testing incluye a todos los testers de todos los grupos de testing. Aunque los gremios son especializados, cualquier persona puede unirse a ellos.

Factor autobús



El "factor autobús" es un indicador de la concentración de conocimiento en cada miembro del equipo.

Imagine que a su equipo lo atropella a todos un autobús, ¿Cuántas personas deberían resultar heridas para que el proyecto se cancele?

Mientras más grande sea el factor autobús, mejor se está compartiendo el conocimiento entre los miembros del equipo, por lo que será más fácil la mejora continua.

Por ejemplo, si el factor autobús fuera muy bajo 1/8 (si falta 1 miembro en el equipo de 8 personas el proyecto no puede avanzar), el proyecto tiene alto riesgo, por lo que hay que mejorar la transferencia del conocimiento entre los miembros de ese equipo para aumentar ese factor.

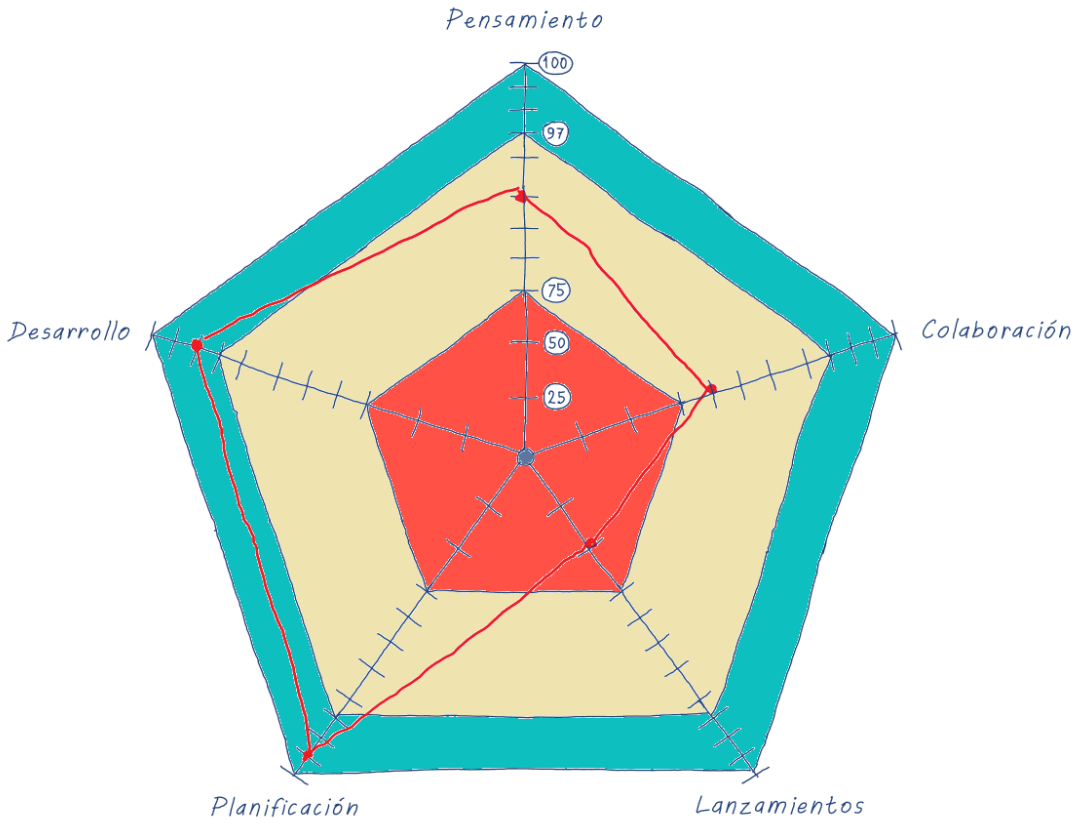
Por otro lado, un factor autobús alto de 6/8 significa que tendrían que verse afectado más de 6 miembros de ese equipo para que el proyecto sea cancelado.

Autoevaluación [31]

La autoevaluación es el proceso en el que un individuo, una organización o un equipo realiza una revisión integral de sí mismos para comprender las fortalezas, debilidades y oportunidades que permitirán mejorar su desarrollo.

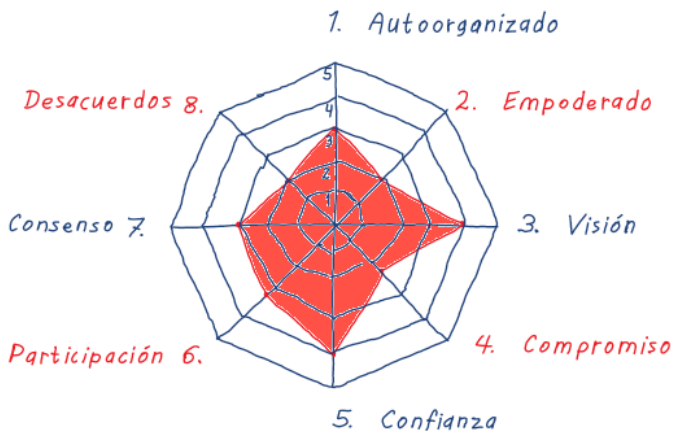
El objetivo principal de la autoevaluación es estimular el aprendizaje y el cambio, así como el entusiasmo por la aplicación de técnicas ágiles.

James Shore ha desarrollado una encuesta de autoevaluación que se enfoca en las prácticas de XP para que los equipos puedan evaluar su rendimiento. La encuesta mide el rendimiento del equipo en una escala 0-100 para 5 categorías.



[31] James Shore, The art of agile development.

Autoevaluación colaborativa [32]



La autoevaluación colaborativa de Jean Tabaka se basa en 8 preguntas que contestan los miembros del equipo con una escala del 1 (totalmente en desacuerdo) hasta 5 (totalmente de acuerdo).

1. **Autoorganizado**. ¿El equipo se autoorganiza frente a la organización de mando y control de arriba hacia abajo?
2. **Empoderado para tomar decisiones**. ¿El equipo está facultado para discutir, evaluar y tomar decisiones en lugar de ser dictado por una autoridad externa?
3. **Creencia en la visión y el éxito**. ¿Los miembros del equipo comprenden la visión y los objetivos del proyecto y realmente creen que, como equipo, pueden resolver cualquier problema para lograrlos?
4. **Equipo comprometido**. ¿Los miembros del equipo están comprometidos a tener éxito como equipo frente al éxito individual a cualquier costo?
5. **Confiar uno en el otro**. ¿El equipo tiene la confianza para trabajar continuamente para mejorar su capacidad de actuar sin miedo, enojo o intimidación?
6. **Toma de decisiones participativa**. ¿El equipo está comprometido en la toma de decisiones participativa versus inclinarse a la toma de decisiones autoritaria o sucumbir a las decisiones de otros?
7. **Impulsado por consenso**. ¿Las decisiones se basan en el consenso y no en el líder?
8. **Desacuerdo constructivo**. ¿El equipo es capaz de negociar a través de una variedad de alternativas e impactos que rodean una decisión y diseñar la que proporcione el mejor resultado?

[32] Jean Tabaka, Collaboration explained.

.

7.5 Ejercicios – Mejora continua

1 Ordene de 1 a 3 los siguientes ciclos de retroalimentación.

Orden	Retroalimentación
	Retrospectiva de la iteración
	Reunión diaria
	Revisión de la iteración

2 Ordene de 1 a 4 el siguiente ciclo de aprendizaje.

Orden	Ciclo
	Aprender
	Desarrollar
	Evaluar
	Planificar

3 Conteste verdadero o falso.

Frase	V	F
El valor de una plataforma de bajo código es que permite el desarrollo iterativo y colaborativo, innovando sin detenerse por los detalles técnicos.		
En cualquier momento, incluso al principio del proceso de desarrollo, el cliente y desarrollador pueden sentarse juntos para discutir y revisar la funcionalidad.		
La innovación ocurre en el punto de equilibrio entre el caos y el orden, entre la flexibilidad y la estabilidad.		
Las aplicaciones de alto código permiten ofrecer retroalimentación inmediata sin necesidad de esperar a que finalice una iteración.		
Las organizaciones ágiles han creado una cultura donde se permite fracasar.		
Mientras antes se fracase, más alto será el costo de ese aprendizaje.		

4 Ordene de 1 a 5 las 5S del Kaizen.

Orden	5S
	Clasificación
	Limpieza
	Mantener la disciplina
	Normalización
	Orden

5 Ordene de 1 al 5 el mapeo del flujo de valor.

Orden	Mapeo del flujo de valor
	Crear un mapa de flujo de valor del proceso actual.
	Crear un nuevo mapa de flujo de valor del estado optimizado.
	Desarrollar una hoja de ruta para lograr el estado futuro.
	Planificar la revisión del proceso para calibrar y optimizar continuamente.
	Revisar el mapa para encontrar retrasos, desperdicios y restricciones.

6 Conteste verdadero o falso.

Frase	V	F
Durante la reunión pre-mortem se realiza un listado de las causas que llevaron al proyecto a su muerte.		
El control empírico del proceso se basa en la transparencia, inspección y colaboración.		
En la reunión de proyecto pre-mortem el equipo imagina que el proyecto fracasó.		
Kanban Kata utiliza preguntas para ayudar a mejorar con cambios de gran escala.		

7 Qué marco y proceso recomienda para cada situación: Ninguno funcionaría, Ágil o Predictivo.

Incertidumbre		Marco / Procesos
Requisitos	Tecnológica	
Alta	Alta	
Media	Media	
Baja	Baja	

8 Ordene de 1 a 3 el ciclo de retroalimentación del producto.

Orden	Retroalimentación
	Aprender
	Construir
	Medir

9 Conteste verdadero o falso con relación al NPS.

Frase	V	F
Los pasivos responden con una calificación de 3		
Si hay un 50% de promotores, 30% de pasivos y 20% de detractores, el NPS es 30%		
NPS es un índice entre 0% y 100%		
Un NPS superior al 50% es excelente		
Los participantes responden en una escala de 1 a 5		
NPS mide la lealtad de los clientes basado en 3 preguntas		

10 Conteste verdadero o falso con relación a las encuestas CSAT y CES.

Frase	V	F
CES se puede utilizar después del primer uso de la nueva funcionalidad		
CES se utiliza para medir la satisfacción en una nueva funcionalidad		
CSAT se utiliza para averiguar si el producto es fácil de usar		
CSAT utiliza una escala del 1 al 10		
CSES utiliza 5 escalas desde muy fácil a muy difícil		

11 ¿Cuál de los siguientes factores aumentan la motivación del equipo?

Factor	
Conectar al equipo con los usuarios	
Empoderar al equipo para que ellos tomen sus propias decisiones	
Pemitir que el equipo sea dueño de una porción del sistema	
Permitir que el equipo sea dueño de su propio proceso	
Reducir el tamaño del equipo tanto como sea posible	

12 Marque qué se obtiene como resultado de la reunión de retrospectiva.

Resultado de la retrospectiva	
Mejora en la calidad del producto	
Mejora en la productividad del equipo	
Plan de mejora del proceso	
Retrospectivas de lanzamiento	
Retrospectivas sorpresa	

13 Ordene de 1 al 5 las fases de una retrospectiva.

Orden	Retrospectiva
	Cierre
	Decidir qué hacer
	Generar información
	Preparación
	Recopilar datos

14 Conteste verdadero o falso con relación a mejorar el equipo.

Frase	V	F
Con “Focus on - focus off” los miembros del equipo se enfoquen en patrones productivos como indagar, diálogo, conversación, entender al otro.		
En el método ESVP, la S es para calificar a los “compradores” que asisten a la reunión a mirar lo que aparece y si algo les gusta, se lo llevan para usar.		
La autoevaluación colaborativa de Jean Tabaka se basa en 8 preguntas		
La encuesta de autoevaluación de James Shore utiliza una escala del 1 al 3		
Mientras más bajo sea el factor autobús, mejor se está compartiendo el conocimiento entre los miembros del equipo.		
ROTI es un método para medir el tiempo dedicado a reuniones de retrospectiva.		
Se suele evaluar la temperatura del equipo recopilando agradecimientos, emociones, quejas, nueva información, enigmas y deseos.		
Un guild es un grupo de personas transversales a la organización que se juntan para compartir conocimientos y buenas prácticas.		
Una herramienta para recopilar las quejas del equipo al finalizar la retrospectiva es la herramienta Plus / Delta.		

Las próximas 3 preguntas son un ejemplo del estilo de preguntas del examen PMI-ACP®. Para practicar con más preguntas situacionales, puede adquirir el simulador PMI-ACP® de Pablo Lledó en www.pablolledo.com.

15 Las historias de usuario de un proyecto se están completando en cada iteración acorde a lo planificado. Sin embargo, Wanda, coordinadora de ese proyecto, les dice a los miembros de su equipo durante una reunión lo siguiente: “Supongan que hace una semana deberíamos haber realizado el primer lanzamiento y todavía no sabemos cuándo ocurrirá ese hito, por lo tanto, la alta gerencia decide cancelar el proyecto”. ¿Por qué estará siendo tan pesimista Wanda?

A. Porque la velocidad del equipo tiene una tendencia decreciente
B. Porque quiere identificar oportunidades del proyecto
C. Porque el cliente no participó en la aceptación formal de las historias completadas
D. Porque está realizando un análisis pre-mortem

16 Durante la reunión de revisión de la iteración el dueño del producto no aprueba los resultados de esa iteración. ¿Porqué podría haber ocurrido esto?

A. El producto incremental no incluye las características del listado de trabajo pendiente

B. El equipo no completó una historia de usuario priorizada para esa iteración

C. El cliente tenía expectativas de que el equipo trabajaría a mayor velocidad

D. La demostración no incluye un producto funcionando

17 Un equipo scrum que necesita acelerar las entregas del proyecto decidió adaptar los procesos eliminando las reuniones de retrospectiva porque las consideran una pérdida de tiempo que no agrega valor. ¿Por qué podrían haber adaptado los procesos de esta forma?

A. La metodología scrum es lo suficientemente flexible para eliminar cualquiera de sus componentes: roles, eventos, artefactos o reglas

B. No es necesario comprender en profundidad las razones detrás de una metodología, ya que la adaptación de procesos a medida de la organización generará mejores resultados

C. En lugar de investigar por qué la retrospectiva no agregaba valor, la solución más fácil fue eliminarla

D. Este proyecto ágil no necesita del proceso de inspección y adaptación

7.6 Respuesta ejercicios – Mejora continua

Retroalimentación

Orden (1º, 2º, 3º)	Retroalimentación
3	Retrospectiva de la iteración
1	Reunión diaria
2	Revisión de la iteración

Ciclo de aprendizaje

Orden	Ciclo
4	Aprender
2	Desarrollar
3	Evaluar
1	Planificar

Retroalimentación continua y Mejorar procesos

Frase	V	F
El valor de una plataforma de bajo código es que permite el desarrollo iterativo y colaborativo, innovando sin detenerse por los detalles técnicos.	√	
En cualquier momento, incluso al principio del proceso de desarrollo, el cliente y desarrollador pueden sentarse juntos para discutir y revisar la funcionalidad.	√	
La innovación ocurre en el punto de equilibrio entre el caos y el orden, entre la flexibilidad y la estabilidad.	√	
Las aplicaciones de alto código permiten ofrecer retroalimentación inmediata sin necesidad de esperar a que finalice una iteración.		√
Las organizaciones ágiles han creado una cultura donde se permite fracasar.	√	
Mientras antes se fracase, más alto será el costo de ese aprendizaje.		√

4R Kaizen

Orden	5S
1	Clasificación
3	Limpieza
5	Mantener la disciplina
4	Normalización
2	Orden

5R Mapeo del flujo de valor

Orden	Mapeo del flujo de valor
1	Crear un mapa de flujo de valor del proceso actual.
3	Crear un nuevo mapa de flujo de valor del estado optimizado.
4	Desarrollar una hoja de ruta para lograr el estado futuro.
5	Planificar la revisión del proceso para calibrar y optimizar continuamente.
2	Revisar el mapa para encontrar retrasos, desperdicios y restricciones.

6R Mejorar procesos

Frase	V	F
Durante la reunión pre-mortem se realiza un listado de las causas que llevaron al proyecto a su muerte.	√	
El control empírico del proceso se basa en la transparencia, inspección y colaboración.		√
En la reunión de proyecto pre-mortem el equipo imagina que el proyecto fracasó.	√	
Kanban Kata utiliza preguntas para ayudar a mejorar con cambios de gran escala.		√

7R Procesos según incertidubmbre.

Incertidumbre		Marco / Procesos
Requisitos	Tecnológica	
Alta	Alta	Caos. Ningún proceso va a funcionar.
Media	Media	Adaptativo. Procesos ágiles.
Baja	Baja	Predictivo. Procesos en cascada.

8R Ciclo de retroalimentación del producto

Orden	Retroalimentación
3	Aprender
1	Construir
2	Medir

9R NPS

Frase	V	F
Los pasivos responden 3		√
Si hay un 50% de promotores, 30% de pasivos y 20% de detractores, el NPS es 30%	√	
NPS es un índice entre 0% y 100%		√
Un NPS superior al 50% es excelente	√	
Los participantes responden en una escala de 1 a 5		√
NPS mide la lealtad de los clientes basado en 3 preguntas		√

10R CES y CSAT

Frase	V	F
CES se puede utilizar después del primer uso de la nueva funcionalidad	√	
CES se utiliza para medir la satisfacción en una nueva funcionalidad		√
CSAT se utiliza para averiguar si el producto es fácil de usar		√
CSAT utiliza una escala del 1 al 10		√
CSES utiliza 5 escalas desde muy fácil a muy difícil	√	

11R Motivación del equipo

Factor	
Conectar al equipo con los usuarios	√
Empoderar al equipo para que ellos tomen sus propias decisiones	√
Pemitir que el equipo sea dueño de una porción del sistema	√
Permitir que el equipo sea dueño de su propio proceso	√
Reducir el tamaño del equipo tanto como sea posible	√

12R Resultado de la retrospectiva.

Resultado de la retrospectiva	
Mejora en la calidad del producto	√
Mejora en la productividad del equipo	√
Plan de mejora del proceso	√
Retrospectivas de lanzamiento	
Retrospectivas sorpresa	

13R Retrospectiva

Orden	Retrospectiva
5	Cierre
4	Decidir qué hacer
3	Generar información
1	Preparación
2	Recopilar datos

14R Mejorar al equipo

Frase	V	F
Con “Focus on - focus off” los miembros del equipo se enfoquen en patrones productivos como indagar, diálogo, conversación, entender al otro.	√	
En el método ESVP, la S es para calificar a los “compradores” que asisten a la reunión a mirar lo que aparece y si algo les gusta, se lo llevan para usar.	√	
La autoevaluación colaborativa de Jean Tabaka se basa en 8 preguntas	√	
La encuesta de autoevaluación de James Shore utiliza una escala del 1 al 3		√
Mientras más bajo sea el factor autobús, mejor se está compartiendo el conocimiento entre los miembros del equipo.		√
ROTI es un método para medir el tiempo dedicado a reuniones de retrospectiva.	√	
Se suele evaluar la temperatura del equipo recopilando agradecimientos, emociones, quejas, nueva información, enigmas y deseos.	√	
Un guild es un grupo de personas transversales a la organización que se juntan para compartir conocimientos y buenas prácticas.	√	
Una herramienta para recopilar las quejas del equipo al finalizar la retrospectiva es la herramienta Plus / Delta.		√

15R Respuesta D. Reunión de proyecto pre-mortem: el equipo imagina que el proyecto ha fracasado y luego trabaja hacia atrás para determinar cuáles podrían haber sido las causas de ese fracaso. El objetivo de la reunión es anticipar los riesgos negativos y actuar para que no lleguen a producirse. Al colocar al equipo en una situación ficticia de fracaso de proyecto, suele ser más simple identificar potenciales riesgos negativos en las fases iniciales del proyecto.

16R Respuesta B. El dueño del producto representa los intereses del cliente y aprobará una iteración cuando el equipo de desarrollo complete todas las historias que se habían priorizado del listado de trabajo pendiente antes de comenzar con esa iteración. / El producto incremental no incluye todas las características del product backlog, sino solamente las características definidas para esta iteración. / El cliente aprueba la iteración si se realizó el trabajo acordado de manera independiente de la velocidad del equipo. / No siempre al final de una iteración el resultado es el de un producto funcionando.

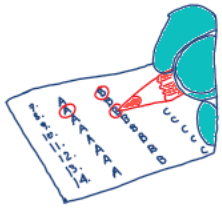
17R Respuesta C. Ken Schwaber, cofundador de Scrum en la Guía de Scrum, dice: "Los roles, eventos, artefactos y reglas de Scrum son inmutables y aunque la implementación de solo partes de Scrum es posible, el resultado no es Scrum". Scrum es una unidad cohesiva donde el todo es mayor que la suma de las partes. En otras palabras, la metodología completa funciona en conjunto y solo puede ser efectiva si todos los componentes individuales se complementan entre sí. / Antes de que el equipo Scrum busque razones para decidir anular las reuniones de retrospectiva, deberían explorar cómo hacer de la retrospectiva una experiencia significativa y valiosa. / Para la adaptación de cualquier marco ágil primero debemos comprender las razones detrás de una práctica específica que se incluye en el marco, antes de decidir que no la necesitamos. Eliminar partes de un marco sin comprender su propósito es como tirar una pieza que no nos gusta de una máquina sin saber lo que hace, y luego esperar que esta máquina siga funcionando correctamente. Quizás si el equipo entendiera y valorara el concepto de inspección y adaptación, necesario en cualquier proyecto ágil, habrían explorado diferentes formas de ejecutar la retrospectiva e intentar hacerla parte integral de su práctica.

Tips de examen

Quienes hayan rendido la certificación PMP® notarán que la preparación para la certificación PMI-ACP® será bastante más fácil. Por lo general, les va a requerir un 50% menos de horas de estudio y tendrán una probabilidad de aprobar en el primer intento muchísimo mayor.

Si bien el examen PMI-ACP® no es tan difícil como el examen PMP®, PMI-ACP® requiere una amplia gama de conocimientos ágiles, familiaridad con los temas y la capacidad de responder preguntas rápidamente.

El examen PMI-ACP no trata solo sobre conocimientos de Scrum, Kanban o cualquier otra metodología ágil, sino de tener buenas habilidades de lectura y comprensión de textos. Por lo tanto, es fundamental practicar con exámenes de simulaciones antes de realizar el examen real.



Tipos de preguntas

- Casi todas las preguntas serán situacionales de corta redacción, por lo que no requieren demasiado tiempo en leerlas.
- Las preguntas se basan en proyectos reales que utilizan ciclos adaptativos, lo que incluye una amplia gama de ejemplos más allá de proyectos de software.
- Algunas preguntas incluirán gráficos o tablas para interpretar esa información.

Contenidos de las preguntas

Aproximadamente el 70% de las preguntas estarán relacionadas con alguno de estos temas:

Análisis de riesgos	Burndown o Burnup chart	Cambios de alcance
Causa raíz del problema	Colaboración con el cliente	Colocación del equipo
Comunicación osmótica	Conciencia social	Defectos escapados
Definir problemas	Desarrollo guiado por pruebas	Deuda técnica
Diagrama de flujo acumulativo	Estimación por afinidad	Gestión de conflictos
Historias de usuario	Holguras XP	Integración continua
Kanban	Lanzamientos	Manifiesto ágil
Modelo Kano	MoSCoW	Planificación de iteración
Póker de planificación	Priorización de historias	Product backlog
Producto viable mínimo	Programación en pares	Radiadores información
Refactorización	Requisitos	Retrospectivas
Reuniones diarias	Scrum	Spikes
Test exploratorio	Tiempo de entrega y de ciclo	Trabajo energizado
Triángulo ágil	Velocidad del equipo	WIP

Consejos de estudio

1º - Leer el PMI-ACP Handbook antes de planificar el examen.

2º - Estudiar la Guía Ágil del PMI®

3º - Leer un Capítulo de este libro y realizar los ejercicios al final del Capítulo.

4º - Realizar el examen del Simulador PMI-ACP de Pablo Lledó relacionado con ese capítulo del libro.

5º - Repetir los pasos 3 y 4 con todos los capítulos de este libro.

6º - Realizar los exámenes parciales y finales del Simulador PMI-ACP de Pablo Lledó simulando un ambiente de examen real sin interrupciones.

7º - Complementar los estudios con otras preguntas de simulación.

¿Cómo saber si estoy en condiciones de rendir y aprobar?

Interpretación de los resultados según los exámenes de simulación:

0% - 50%	Estudiar bastante más	
50% - 70%	Anotarse para rendir y seguir estudiando	
70% - 100%	Listo para rendir	

Si tus calificaciones superan el 50% deberías anotarte para rendir el examen. Si esperas a estar totalmente preparado, nunca te vas a anotar ya que siempre vas a creer que no estás listo.

Si en los exámenes de simulación obtienes calificaciones superiores al 70% al primer intento, o sea sin repetir las mismas preguntas, deberías rendir el examen lo antes posible para que no te olvides de esos conocimientos.

Recomendaciones finales

- Responder según la Guía del PMI-ACP®, no tu experiencia en proyecto ágiles.
- LEER SIEMPRE LAS 4 OPCIONES antes de contestar una pregunta.
- No amargarse con las preguntas complejas, marcar cualquier letra. No dejes nunca una pregunta sin tildar alguna respuesta. Además, marcas esa pregunta para revisión y vuelves a revisarla antes de entregar tu examen.
- Si estudias bien, sólo tendrás dudas en el 20% de las preguntas. Puedes volver al final del examen a revisar sólo esas preguntas ya que estarás más tranquilo porque el restante 80% lo marcaste sin dudar.
- No estudiar la noche anterior. Necesitas descansar muy bien para rendir durante 3 horas. Tu cerebro necesita procesar todo lo aprendido y agregar más conceptos la noche anterior puede ser perjudicial.
- Por último, subes tu comentario en nuestro blog de que ya aprobaste https://pablolledoblog.wordpress.com/ para que publiquemos tu nombre en nuestra newsletter PM1p.

La disrupción tecnológica, inteligencia artificial, trabajo remoto y los equipos virtuales han cambiado la forma de gestionar nuestros proyectos. Por su parte, la nueva economía está indicando que las empresas exitosas serán sólo aquellas que hagan realidad su estrategia mediante proyectos eficientes que generen los beneficios esperados.

En este contexto ya no alcanza con conocer herramientas para gestionar proyectos de ciclos predictivos con procesos en cascada, sino que cualquier profesional de proyectos debe conocer también cuáles son las metodologías y buenas prácticas para gestionar proyectos con ciclos adaptativos que requieren entregar valor al cliente de manera incremental y con mayor celeridad que en el pasado.

Este libro en formato de "*apuntes de clase*" fue diseñado para quienes quieran preparar la certificación internacional PMI-ACP® o para quienes necesiten mejorar sus conocimientos para la dirección de proyectos.

Las ventajas de estudiar de este libro son:

- Obtener una visión de los principios ágiles más utilizados.
- Interactuar con miembros del equipo de proyectos ágiles.
- Quedar a poca distancia de aprobar el examen PMI-ACP®.
- Ser mejores profesionales de proyectos.

Pablo Lledó
PMI-ACP, PMP

978-0-9995073-2-2

pablolledó
projectManagement